

# Interactive ontology matching based on partial reference alignment

Xue, Xingsi; Yao, Xin

DOI:

[10.1016/j.asoc.2018.08.003](https://doi.org/10.1016/j.asoc.2018.08.003)

License:

Creative Commons: Attribution-NonCommercial-NoDerivs (CC BY-NC-ND)

*Document Version*

Peer reviewed version

*Citation for published version (Harvard):*

Xue, X & Yao, X 2018, 'Interactive ontology matching based on partial reference alignment', *Applied Soft Computing*, vol. 72, pp. 355-370. <https://doi.org/10.1016/j.asoc.2018.08.003>

[Link to publication on Research at Birmingham portal](#)

## General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

## Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact [UBIRA@lists.bham.ac.uk](mailto:UBIRA@lists.bham.ac.uk) providing details and we will remove access to the work immediately and investigate.

## Accepted Manuscript

Title: Interactive Ontology Matching based on Partial Reference Alignment

Author: Xingsi Xue Xin Yao

PII: S1568-4946(18)30451-4

DOI: <https://doi.org/doi:10.1016/j.asoc.2018.08.003>

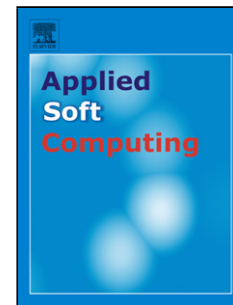
Reference: ASOC 5031

To appear in: *Applied Soft Computing*

Received date: 25-12-2016

Revised date: 9-6-2018

Accepted date: 2-8-2018



Please cite this article as: Xingsi Xue, Xin Yao, Interactive Ontology Matching based on Partial Reference Alignment, *Applied Soft Computing Journal* (2018), <https://doi.org/10.1016/j.asoc.2018.08.003>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

## Highlights

In this paper, we propose an interactive ontology matching based on Partial Reference Alignment (PRA) to efficiently match the heterogeneous ontologies. Our major contributions are as follows:

- (1) A PRA-based ICHEA is proposed to adaptively determine the timing of getting user involved, and automatically search for potential mappings;
- (2) Three techniques, i.e. partition of the ontologies into similar segment pairs, determination of problematic mappings, and automatic validation of multiple conflicting mappings, are proposed to reduce user workload;
- (3) An asymmetrical profile-based similarity measurement and a mapping confidence propagation are proposed to increase the value of user involvement, and, at the same time, reduce the negative effect brought by erroneous user validations.

# Interactive Ontology Matching based on Partial Reference Alignment

Xingsi Xue<sup>a,b,c</sup> and Xin Yao<sup>d</sup>

<sup>a</sup> *College of Information Science and Engineering, Fujian University of Technology, Fuzhou, Fujian, China*

<sup>b</sup> *Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian, China*

<sup>c</sup> *Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou, Fujian, China*

<sup>d</sup> *CERCIA, School of Computer Science, University of Birmingham, Edgbaston, Birmingham, U.K.*

---

## Abstract

The technique that enables the user and the automatic ontology matching tool to cooperate with each other to generate high-quality alignments in a reasonable amount of time is referred to as the interactive ontology matching. Interactive ontology matching poses a new challenge in a way of how to efficiently leverage user validation to improve the ontology alignment. To address this challenge, this paper presents an innovative interactive ontology matching technique based on Partial Reference Alignment (PRA) to better balance between the large workload posed on users and the demand of improving the quality of ontology alignment. In particular, a PRA-based Interactive Compact Hybrid Evolutionary Algorithm (ICHEA) is proposed to reduce user workload, by adaptively determining the timing of involving users, showing them the most problematic mappings, and helping them to deal with multiple conflicting mappings simultaneously. Meanwhile, it increases the value of user involvement by propagating the confidences of validated mappings, as well as reducing the negative effects brought by the erroneous user validations. The well-known OAEI 2016's benchmark track and interactive track are utilized to test the performance of this approach. The experimental results on benchmark track show that both the f-measure and the f-measure per second of this approach outperform those of the OAEI

participants and three state-of-the-art Evolutionary Algorithm (EA) based ontology matching techniques. In addition, the experimental results of three interactive testing cases further show that ICHEA can efficiently determine high-quality ontology alignments under different cases of user error rates, and the performance of the approach is generally better than that of state-of-the-art interactive ontology matching systems.

*Keywords:* interactive ontology matching, partial ontology alignment, interactive compact hybrid evolutionary algorithm.

---

## 1. Introduction

Although ontology is reckoned as a solution to data heterogeneity on the Semantic Web, the subjectivity of different ontology designers leads to the generation of heterogeneous ontologies. In order to support the semantic inter-operability in many domains through disparate ontologies, it's necessary to identify the correspondences out of semantically identical entities inside two heterogeneous ontologies, which is commonly known as ontology matching [1]. Since manual identification of semantic correspondences is extremely impractical, especially that the ontologies could contain hundreds even thousands of entities, many automatic ontology matching tools have been proposed in recent years. However, due to the complexity in the process of ontology matching, ontology alignments generated by automatic matching tools should be validated by users to guarantee the quality [2]. The technique that enables users and automatic tools to cooperate with each other to generate a high-quality alignment in a reasonable amount of time is referred to as interactive ontology matching [3].

The ontology can be described through its architecture graph (the nodes represent concepts and instances, while the edges stand for the relationship between them). Solving ontology matching problem is the process of determining the largest isomorphic subgraph out of the two architecture graphs of two ontologies to be matched. Since modeling ontology matching is a complex (nonlinear with many local optimal solutions) and time-consuming task (large scale), particularly when the number of ontology entities is significantly large. Evolutionary Algorithm (EA) could be an efficient approach to address this problem. Comparing with other evolutionary methods such as Particle Swarm Optimization (PSO) algorithm [4], Hybrid Evolutionary Algorithm (HEA) [5, 6] can better balance the global search and local search,

and work better in reducing the possibility of premature convergence whereas increasing the convergence speed. Therefore, it is able to solve the problem of ontology matching in a more efficient way. Actually, HEA-based ontology matching techniques perform better in determining ontology alignments than state-of-the-art ontology matching systems do [7, 8, 9]. However, since it is necessary to execute the ontology matching within a specific run time, aparting from the quality of alignments, the execution time and main memory consumption are essential too. According to [10], if properly designed, a population-based algorithm with a very small population size can efficiently solve the large-scale problem. As a result, a new category of EA emerges, which is the so-called Compact EA (CEA). It employs the search logic of population-based algorithms without storing or processing the entire population. On the contrary, it utilizes probabilistic representatives of the population to perform the optimized process [11]. Thus, CEA-based ontology matching technique greatly improves the efficiency of traditional EA-based approaches. In addition, traditional EA-based ontology matching techniques usually require the domain expert to provide a reference alignment to evaluate the quality of the obtained ontology alignments. Since the number of possible correspondences grows quadratically over the number of entities inside the ontology, the typical approach is not feasible for large-scale matching tasks. To overcome this drawback, Partial Reference Alignment (PRA), which is a set of example mappings that can be provided by a domain expert in a reasonable amount of time, is proposed to replace the reference alignment to evaluate the quality of ontology alignment [12, 13]. Nevertheless, the research on it is still in its infancy.

How to minimize user workload, while on the other hand, maximizing the value of user involvement is one of the main challenges in interactive ontology matching domain [14]. To address this challenge, this paper proposes a PRA-based Interactive Compact Hybrid Evolutionary Algorithm (ICHEA) to better balance between user workload and the demand of improving the quality of ontology alignment. Firstly, the timing of getting user involved through CHEA is determined adaptively. During each validating process, the most problematic mappings for validation are presented to the user, and help him to validate multiple conflicting mappings simultaneously. After that, we propagate the confidences from validated mappings to their neighborhood, which manages to reduce the negative effects brought by erroneous user validations. In particular, the major contributions of this work lie in the following aspects:

- A PRA-based ICHEA is proposed to adaptively determine the timing of getting user involved, and automatically search for potential mappings;
- Three techniques, i.e. partition of the ontologies into similar segment pairs, determination of problematic mappings, and automatic validation of multiple conflicting mappings, are proposed to reduce user workload;
- An asymmetrical profile-based similarity measurement and a mapping confidence propagation are proposed to increase the value of user involvement, and, at the same time, reduce the negative effect brought by erroneous user validations.

The rest of the paper is structured as follows: Section 2 describes relevant work of this paper; Section 3 presents the evaluation of PRA-based ontology alignment; Section 4 introduces the framework of PRA-based interactive ontology matching; Section 5 illustrates the automatic ontology matching process which adaptively determines the timing of involving users; Section 6 demonstrates three techniques to reduce user workload; Section 7 explains the approach of increasing the value of user involvement; Section 8 sketches the outline of ICHEA; Section 9 unfolds the studies and analysis of the experiments; last but not least, Section 10 draws the conclusion and forecasts the future work.

## 2. Related Work

### 2.1. Interactive Ontology Matching

The performance of automatic ontology matching techniques is quite limited, since adopting more advanced alignment techniques brought diminishing returns. The reason is the complexity and intricacy of the ontology alignment process, as well as that each task has its own uniqueness, dictated both by the domain and the design of the ontologies. Therefore, automatic generation of mappings should be regarded only as the first step towards the final alignment, with the pre-requisite of making user validation an essential step to guarantee the alignment quality [3]. For this reason, in recent years, many interactive ontology matching techniques have been proposed to improve the quality of ontology alignment by exploiting the user's interventions. Existing interactive ontology matching techniques can be categorized

according to different phases of involving users, i.e., before, during or after the automatic matching phase.

AML [15] starts interacting with the user in selecting and repairing phases, which is after the automatic matching phase. The user's inputs are employed to filter the mappings contained in the final alignment. Alin [3] also gets user involved after the automatic matching phase. It generates an initial set of candidate mappings between classes based on six string-similarity metrics and a stable-marriage algorithm. If a candidate mapping generates the maximum similarity value for all six metrics, then it is added to the final alignment. The rest are sorted by the sum of the metrics, and will be presented to the user one by one. The candidate mapping will be transferred to the final alignment, once the user accepts it. Data and object correspondences related to the accepted mapping are then added to the candidate mapping list, while all candidate mappings that are part of alignment anti-patterns with the approved mapping will be removed from the list of candidates. The process continues until no more candidate mappings remain. Before automatic matching phase, LogMap [16] interacts with users to generate candidate mappings, and then it further employs lexical, structure and reasoning-based techniques to discard some of the mappings. Finally, it requires the user to validate those mappings that are not clear-cut, and then determines the final alignment. XMap [17] applies two thresholds to filter candidate mappings for user validation. One is for mappings that are directly added to the final alignment; the other is for those presented to the user for validation. High threshold is set for the latter to minimize the number of requests. The rejected candidate mappings from the oracle and the requests are mainly incorrect mappings. The mappings accepted by the user will be moved to the final alignment. ServOMBI [18] relies on terminological indexing strategy provided by the ServO Ontology Repository (OR) system [19] to reduce the search space, and compute an initial set of candidate mappings based on the terminological description of the ontology entities. After that, it utilizes a machine learning approach and a modified contextual similarity measure to determine the ontology alignment. In the post-matching phase, the user is asked to validate candidate mappings to determine the final alignment.

These state-of-the-art interactive ontology matching techniques depend exclusively on user interactions before or after the matching to distinguish their candidate mappings. While our approach mainly focuses on the iteratively validating process, where the user gets involved during the iterative automatic matching process. In each iteration, the alignment generated from



the previous one is improved in terms of completeness (whether all correct correspondences have been determined) and soundness (whether all determined correspondences are correct correspondences), which effectively raises the quality of the ontology alignment.

## 2.2. *Ontology Matching based on Partial Reference Alignment*

In recent years, some ontology matching systems have begun to utilize the PRA to guide the matching process. Lambrix et al. [12] was the first to use PRA to partition ontologies, and determine the appropriate weights among various ontology similarity measures. They further presented an ontology matching framework that uses PRA to generate mapping suggestions for user validation [20]. Duan et al. [21] utilized the PRA to evaluate the ontology alignment obtained in an iterative supervised-learning process, which was also dedicated to determine the optimal weights to combine various ontology matchers for matching ontology entities. Similarly, Tan et al. [22] used PRA to implement an ontology alignment recommendation approach, which firstly used PRA to evaluate the ontology alignments, and then provided recommendations based on evaluation results. ECOMatch [23] utilized PRA to fine tune its parameter set, which was then used to match the ontologies. More recently, Xue et al. [13] proposed a novel PRA constructing approach based on ontology clustering algorithm, where EA was used to determine the optimal weights to aggregate different ontology matcher's alignment.

All of the above methods use PRA as a static input to reduce the search space of the algorithm or to tune the system's parameters, while in our work, during the evolving process, PRA can be updated after each user's interaction. The updated PRA will better evaluate the quality of the alignment and guide the search direction of ICHEA, and as a result of which, the performance of the algorithm is improved.

## 3. **Partial Reference Alignment based Ontology Alignment Evaluation**

In this study, an ontology  $O$  is defined as 5-tuple  $(C, P, I, \Lambda, \Gamma)$  [7], where  $C, P, I, \Lambda, \Gamma$  are referred to the set of classes, properties, instances, axioms and annotations respectively. In addition, an ontology alignment  $A$  between two ontologies is a correspondence set. Each correspondence is a 4-tuple  $(e, e', n, r)$ , where  $e$  and  $e'$  are the entities of two ontologies,  $n \in [0, 1]$  is a

confidence value holding for the correspondence between  $e$  and  $e'$ , and  $r$  is the relationship between  $e$  and  $e'$ , which refers to equivalence in this work.

In Figure 1, there are two simple ontologies  $O_1$  and  $O_2$ , and an alignment between them. Classes are shown in rectangles with rounded corners, e.g., in  $O_1$ , Book is specialization (subclass) of Product, but their relationships are shown without the latter. It is the same with price being an attribute defined on the integer domain and creator being a property. Albert Camus: Lachute is a shared instance. Correspondences are shown as thick arrows that link an entity from  $O_1$  with an entity from  $O_2$ . They are annotated with the relationship that is expressed by the correspondence: for example, Person in  $O_1$  is less general ( $\sqsubseteq$ ) than Human in  $O_2$ . Assume that an e-commerce company acquires another one. Technically, this acquisition requires the integration of their information sources, and hence, of the ontologies of these companies. The documents or instance data of both companies are stored according to ontologies  $O_1$  and  $O_2$ , respectively. In this example, these ontologies contain subsumption statements, property specifications and instance descriptions. The first step in integrating ontologies is matching, which discovers correspondences, namely candidate entities to be merged or to have subsumption relationships under an integrated ontology. Once the correspondences between two ontologies have been discovered, they can be used, for instance, for generating query expressions that automatically translate instances of these ontologies under an integrated ontology. For example, the attributes with labels title in  $O_1$  and in  $O_2$  are the candidates to be merged, while the class with label Monograph in  $O_2$  should be subsumed by the class Product in  $O_1$ .

To evaluate the quality of an alignment and the effectiveness of a matching approach, it is necessary to determine whether all correct correspondences have been discovered (completeness) and whether all discovered correspondences are correct (soundness). Normally, the alignment is assessed in terms of two measures, commonly known as precision and recall. Precision (or soundness) measures the fraction of the selected correspondences that are actually correct. Recall (or completeness) measures the fraction of the number of correct mappings discovered against the total number of existing correct alignments. Maximum precision (no false positive) and maximum recall (no false negative) refer to the absence of type I and type II errors <sup>1</sup> respectively.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Type\\_I\\_and\\_type\\_II\\_errors](https://en.wikipedia.org/wiki/Type_I_and_type_II_errors)

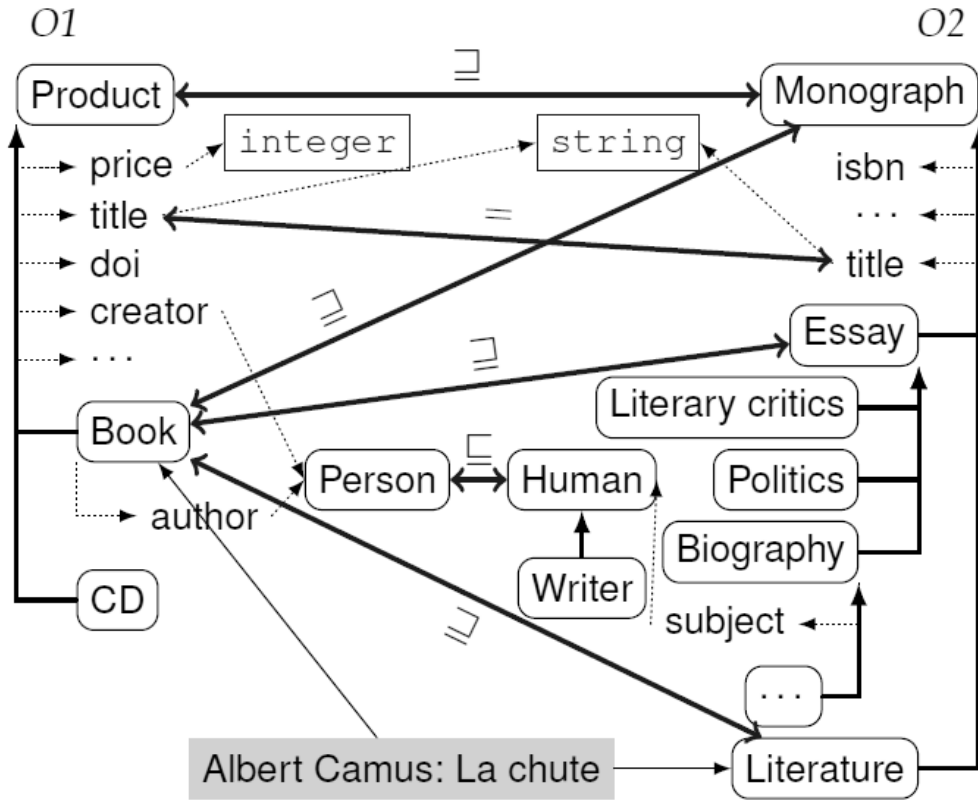


Figure 1: Two simple ontologies and an alignment

Although a precision of 1 means that all correspondences found are correct, it does not imply that all correct ones have been found. Analogously, a recall of 1 means that all the correct correspondences have been discovered, but it does not provide the information about the number of falsely identified ones. Therefore, precision and recall are often balanced against each other by the so-called f-measure, which is the uniformly weighted harmonic meaning of recall and precision. Since f-measure can better balance the precision and recall, it is the most popular indicator that is utilized to measure the quality of an ontology alignment. Given a Reference Alignment (RA)  $R$ , which is the golden ontology alignment provided by the expert, and an alignment  $A$ , *recall*, *precision* and *f-measure* can be defined as follows [24]:

$$precision = \frac{|R \cap A|}{|A|} \quad (1)$$

$$recall = \frac{|R \cap A|}{|R|} \quad (2)$$

$$f - measure = 2 \times \frac{precision \times recall}{precision + recall} \quad (3)$$

Although recall, precision and f-measure can reflect the quality of the resulting alignment, they require domain experts to provide the reference alignment in advance. However, this perfect matching result is generally unknown for difficult real-life matching problems, especially for the large-scale ontology matching task. In this study, the Partial Reference Alignment (PRA) based metrics are proposed to approximately evaluate the quality of ontology alignment. PRA is a subset of RA, which can be provided by a domain expert in a reasonable amount of time. Given an ontology alignment  $A$  and a partial reference alignment  $PRA$ , a partial alignment  $A_p$  is the subset of  $A$  which contains all elements in  $A$  and shares at least one entity with an element in  $PRA$  [13]:

$$A_p = \{(e_1, e_2, n, =) \in A \mid \exists e'_1, n' : (e'_1, e_2, n', =) \in PRA\} \cup \{(e_1, e_2, n, =) \in A \mid \exists e'_2, n' : (e_1, e'_2, n', =) \in PRA\} \quad (4)$$

Through referring to the definitions of recall, precision and f-measure, this paper defines recall, precision and f-measure on PRA by replacing  $R$  and  $A$  with  $PRA$  and  $A_p$  respectively in the formulas.

#### 4. Partial Reference Alignment based Interactive Ontology Matching Framework

According to the statement made by the OAEI organizers, automatic generation of mappings is only the first step towards the final alignment, so the user interaction in the matching process is essential [25]. Therefore, this paper presents a PRA-based interactive ontology matching framework, which is shown in Figure 2.

In this figure, three working phases, i.e., PRA initialization, PRA-based ICHEA, and segment alignment aggregation and evaluation, are outlined

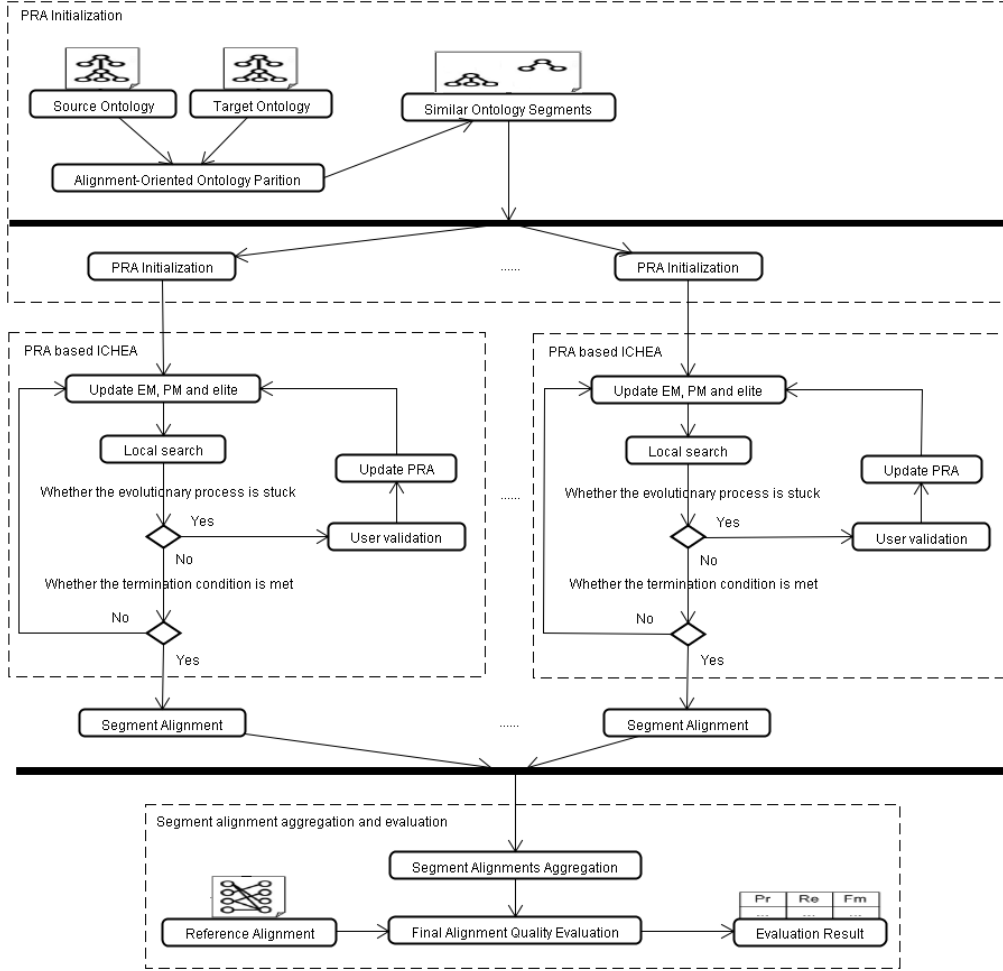


Figure 2: Outline of the Interactive Compact Hybrid Evolutionary Algorithm based Ontology Matching Framework

by dotted-line boxes. A dotted-line box with a rectangle inside represents a working step. A rectangle with a picture indicates the input or output data (e.g., the ontologies, ontology segments or the evaluation results). Two black lines in parallel refer to the beginning and end of two parallel executions respectively. Specifically, the description of three working phases is given as follows:

- PRA initialization: an ontology partition algorithm is used to partition two ontologies into several similar segment pairs. Before matching

each segment pair, the high-confidence segment entity mappings are presented to the user for validation to initialize the PRA of this matching task, i.e., each segment matching task has one particular PRA of its own. The matching process hereafter only needed to be executed on much smaller ontology segments, which reduces the search space of ICHEA and improves its performance;

- PRA-based ICHEA: the PRA-based ICHEA is used to match the ontology segments. When the evolving process gets stuck, the algorithm will get the user involved to guide the algorithm's search direction by introducing new correspondences and filtering error ones;
- Segment alignment aggregation and evaluation: all the segment alignments are aggregated through a greedy strategy, where among all corresponding correspondences from each source ontology entity, the ones with the highest confidence value are selected to form the final ontology alignment. The final ontology alignment is further evaluated through f-measure to compare with the results from other ontology matching techniques.

## 5. Compact Hybrid Evolutionary Algorithm

The slow convergence and premature convergence are two main disadvantages of classical EAs [26], which make these algorithms incapable of effectively searching the optimal solution for ontology matching problem. The Hybrid Evolutionary Algorithm (HEA) is a population-based optimized algorithm with an evolutionary framework which contains and launches local search components in each generation. This marriage between global and local search is helpful to reduce the possibility of the premature convergence, and increase the convergence speed. However, it could be plagued by hardware limitations, and be inadequate in terms of efficiency when facing large-scale tasks. Therefore, the compact version of HEA (CHEA) can be more efficient to solve the ontology matching problem, which simulates the behavior of a population-based HEA by employing, instead of a population of solutions, the probabilistic representation of the population. In addition, since incorporating human directly into the evolutionary cycle can highly improve the search ability of EA [27], an interactive version of CHEA (ICHEA) is proposed, which can utilize the user's knowledge to enhance CHEA's exploration and exploitation.

First, the optimal model of ontology matching problem is presented as follows:

$$\begin{cases} \max & f - \text{measure}_{PRA}(X) \\ \text{s.t.} & X = (x_1, x_2, \dots, x_{|O_1|})^T \\ & x_i \in \{1, 2, \dots, |O_2|\}, i = 1, 2, \dots, |O_1| \end{cases} \quad (5)$$

where  $|O_1|$  and  $|O_2|$  refer to the cardinalities of two ontologies  $O_1$  and  $O_2$  respectively,  $x_i, i = 1, 2, \dots, |O_1|$  is the  $i$ -th pair of correspondence, and the objective is to maximize f-measure on PRA.

Then, three key components of ICHEA are illustrated, i.e., entity and probability matrix, elitism strategy, and local search.

### 5.1. Entity and Probability Matrix

An ontology can be described as an architecture graph (the nodes represent the concepts and instances, and the edges stand for the relationships between them). Solving the problem of ontology matching is the process of determining the largest isomorphic subgraph between two architecture graphs of two heterogeneous ontologies. On this basis, the ontology matching problem can be modeled as a concrete optimization problem with the real value encoding mechanism. Particularly, given a source ontology segment  $seg_{src}$ , and a target ontology segment  $seg_{tgt}$ ,  $|seg_{src}|$  and  $|seg_{tgt}|$  respectively represent the cardinality of each of their own entity set. In this work, an individual's gene length is equal to  $|seg_{src}|$ , and each gene's value is in  $\{-1, 0, 1, \dots, |seg_{tgt}|\}$ . An example of an individual's chromosome representation is given in Figure 3. In particular, the value of -1 (or 0) means its matched entity is undetermined (or none).

The traditional CEA uses Probability Vector (PV) to represent the population [10], which is designed for continuous optimization problem with binary-encoding mechanism. In our work, we modeled the ontology matching problem as a concrete optimization problem with the real value encoding mechanism. Therefore, traditional PV is not suitable for our encoding mechanism. Moreover, Entity Matrix (EM) and Probability Matrix (PM) are proposed to implement the concrete decimal-encoding mechanism. EM and PM respectively restore the information of three solutions, i.e. the elite solution, local best solution and global best solution, and the probabilities of mappings in their corresponding alignments. Elite solution is currently found the best solution by ICHEA. The local best solution consists of the best correspondences that are found by ICHEA for each  $seg_{src}$ 's

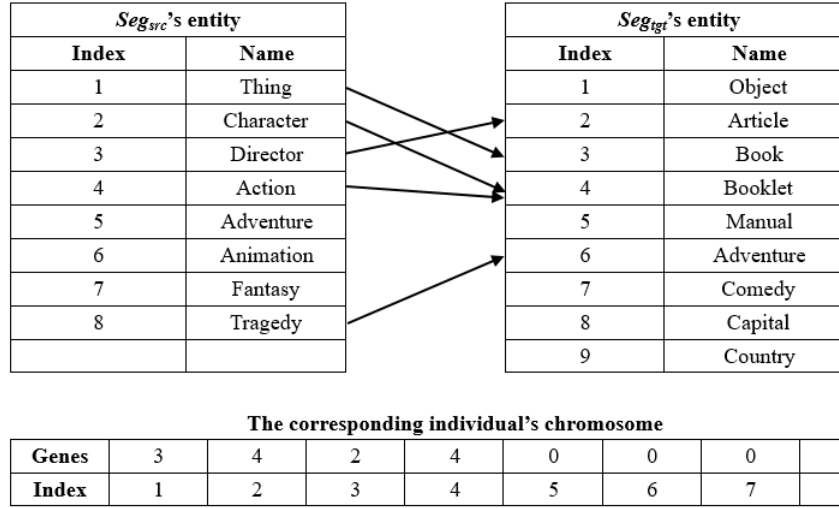


Figure 3: A example of an individual's chromosome representation

entity. The global best solution consists of the mappings that have been validated by the user. Formally,  $EM = [EV_1, EV_2, \dots, EV_{|seg_{src}|}]$  where the  $i$ -th Entity Vector (EV)  $EV_i = (X_i^{elite}, X_i^{localBest}, X_i^{globalBest})^T$ ,  $X_i^{elite}$ ,  $X_i^{localBest}$  and  $X_i^{globalBest}$  are respectively elite's  $i$ -th gene's value, the found best  $i$ -th gene's value by ICHEA and user validated  $i$ -th gene's value. Accordingly,  $PM = [PV_1, PV_2, \dots, PV_{|seg_{src}|}]$  where the  $i$ -th Probability Vector (PV)  $PV_i = (P_i^{elite}, P_i^{localBest}, P_i^{globalBest})^T$ ,  $P_i^{elite}$ ,  $P_i^{localBest}$  and  $P_i^{globalBest}$  respectively represent the probabilities of a new individual's  $i$ -th gene value being the value of  $X_i^{elite}$ ,  $X_i^{localBest}$  and  $X_i^{globalBest}$ .

In the following part, an example is given about generating a new individual through EM and PM. Given  $EM = \begin{bmatrix} 2 & 9 & 8 \\ 5 & 0 & 8 \\ 4 & 1 & -1 \end{bmatrix}$ ,  $PM = \begin{bmatrix} 0.6 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.1 \\ 0.3 & 0.2 & 0.8 \end{bmatrix}$ , where  $P_i^{elite}$ ,  $P_i^{localBest}$  and  $P_i^{globalBest}$ ,  $i = 1, 2, 3$ , the interval  $[0, 1]$  is divided into three sub-intervals:  $[0, P_i^{elite})$ ,  $[P_i^{elite}, P_i^{elite} + P_i^{localBest})$  and  $[P_i^{elite} + P_i^{localBest}, 1]$ . For example,  $P_1^{elite} = 0.6$ ,  $P_1^{localBest} = 0.1$  and  $P_1^{globalBest} = 0.3$  divides the interval  $[0, 1]$  into three sub-intervals  $[0, 0.6)$ ,  $[0.6, 0.7)$  and  $[0.7, 1]$ . Sequentially, three random numbers are generated, such as 0.3, 0.6 and 0.4, which is equal to the column number of EM or PM. Since  $0.3 \in [0, 0.6]$  (the first sub-interval), the new generated individual's first bit value is 2.



Similarly, the second and third bit values are 0 and -1 respectively, i.e., the first entity in  $seg_{src}$  is matched with second entity in  $seg_{tgt}$ , and the second entity in  $seg_{src}$  is matched with none, while the third entity in  $seg_{src}$  is not determined yet, whose value could be in  $\{0, 1, \dots, |seg_{tgt}|\}$ . By repeating the work of generating different random numbers in the above procedure, various individuals can be generated.

### 5.2. Elitism Strategy

In each generation, EM, PM and the best individual found until the current generation, the so-called elite, will be updated. For the sake of clarity, the pseudo-code is shown in Algorithm 1. When the value of  $P_i^{elite}$ ,  $P_i^{localBest}$  or  $P_i^{globalBest}$ ,  $i = 1, 2, \dots, |seg_1|$  is smaller than 0 (or larger than 1), we set it as 0 (or 1).

It is obvious that when all the elements' values in EM are close to 1 or 0, the individuals generated by EM are of high similarity, and the algorithm is exploitative. When the elements' values in EM are around 0.5, however, the individuals generated by EM are of great difference, and the algorithm is explorative. Therefore, the larger  $st$  is, the faster the values of EM's elements will approach 1 or 0, and the algorithm will put more emphasis on exploitation. Otherwise, the emphasis will be on exploration. In order to balance between the exploitation and exploration, in this work, we empirically set  $st = 0.01$  to achieve the highest average alignment quality on all testing cases.

### 5.3. Local Search

In this paper, the local search strategy dedicates to generate various individuals to search the vicinity range of *elite*. To this end, a  $C \times D$  matrix  $M$  is constructed and used to implement the uniform crossover between *elite* and an individual to generate the neighbor. Here,  $C$  is the scale of neighbor population, i.e., the number of generated neighborhood, and  $D$  equals to the length of an individual. With respect to  $C$ , a larger value of it may perform better exploitation, especially for the multi-modal problem, but it will increase the computational complexity. Given a source segment  $Seg_{src}$  and a target segment  $Seg_{tgt}$ , this work empirically sets  $C = 5 + \lceil \max\{|Seg_{src}|, |Seg_{tgt}|\} / 10 \rceil$ , where  $|Seg_{src}|$  and  $|Seg_{tgt}|$  are the number of entities in  $Seg_{src}$  and  $Seg_{tgt}$ , respectively.

**Algorithm 1** Elitism Strategy

**Input:** new solution  $solution^{new}$ , elite solution  $elite$ , entity matrix  $EM$ , probability matrix  $PM$ , step size  $st$ ;

**Output:** Updated  $EM$  and  $PM$ .

```

[winner, loser] = compete( $solution^{new}$ ,  $elite$ );
if  $winner == solution^{new}$  then
  for  $i = 0; i < solution^{new}.length; i++$  do
    if  $solution_i^{new} \neq elite_i$  then
       $X_i^{elite} = solution_i^{new}$ ;
       $P_i^{elite} + st$ ;
       $P_i^{localBest} - st/2$ ;
       $P_i^{localBest} - st/2$ ;
    end if
    if  $solution_i^{new} == X_i^{localBest}$  or  $solution_i^{new}$  is better than  $X_i^{localBest}$ 
    then
       $X_i^{localBest} = solution_i^{new}$ ;
       $P_i^{localBest} + st$ ;
       $P_i^{elite} - st/2$ ;
       $P_i^{globalBest} - st/2$ ;
    end if
    if  $solution_i^{new} == X_i^{globalBest}$  then
       $P_i^{globalBest} + st$ ;
       $P_i^{elite} - st/2$ ;
       $P_i^{localBest} - st/2$ ;
    end if
  end for
else
  if  $winner == elite$  then
    for  $i = 0; i < elite.length; i++$  do
      if  $elite_i \neq solution_i^{new}$  then
         $P_i^{elite} + st$ ;
         $P_i^{localBest} - st/2$ ;
         $P_i^{globalBest} - st/2$ ;
      end if
      if  $elite_i == solution_i^{new}$  then
         $P_i^{elite} - st$ ;
         $P_i^{localBest} + st/2$ ;
         $P_i^{globalBest} + st/2$ ;
      end if
    end for
  end if
end if

```

For the sake of clarity, given a permutation probability  $p_p$  that is normally set to 0.5, an  $M$  can be constructed to implement the uniform crossover operation. The pseudo-code of generating  $M$  is presented as follows:

```
//Initialize  $M$ 
1. for(int  $i = 0; i < C; i++$ )
2.   for(int  $j = 0; j < D; j++$ )
3.      $M_{ij} = 0$ ;
4.   end for
5. end for

//Permutate  $M$ 
6. for(int  $i = 0; i < C; i++$ )
7.   generate  $j = \text{round}(\text{rand}(0, D))$ ;
8.   while ( $\text{rand}(0, 1) < p_p$ )
9.     if ( $j == D$ )
10.       $j = 0$ ;
11.    end if
12.     $M_{ij} = 1$ ;
13.     $j = j + 1$ ;
14.  end while
15. end for
```

$\overline{M}$  can be obtained by swapping the value in  $M$ , i.e., converting the zero elements in  $M$  into one and non-zero elements into zero. On this basis, the neighbor population of the *elite* can be generated through the following formula:

$$\overrightarrow{\text{solution}_{neighbor}} = M \otimes \overrightarrow{\text{elite}} + \overline{M} \otimes \overrightarrow{\text{Solution}} \quad (6)$$

where  $\overrightarrow{\text{elite}} = \begin{bmatrix} \text{elite} \\ \text{elite} \\ \dots \\ \text{elite} \end{bmatrix}_{C \times D}$ ,  $\overrightarrow{\text{Solution}} = \begin{bmatrix} \text{solution}^1 \\ \text{solution}^2 \\ \dots \\ \text{solution}^C \end{bmatrix}_{C \times D}$  and  $\text{solution}^i$ ,

$i = 1, 2, \dots, C$ , is generated by EM and PM, and the operator  $\otimes$  denotes the element-wise multiplication of corresponding matrix elements. Then, the best individual is selected from the neighbor population  $\text{solution}_{neighbor}^i$ ,  $i = 1, 2, \dots, C$ . If it improves the *elite*, the process is repeated until no further improvements can be made.

## 6. Reduce User's Workload

### 6.1. Ontology Partition

Partitioning the large-scale ontology into various segments, where the term “segment” is referred to as a fragment of an ontology, is an efficient way of reducing the algorithm's search space and decreasing unnecessary workload for the user [28]. In this work, an alignment-oriented ontology partition technique is introduced to partition the ontologies into various similar ontology segment pairs. First of all, the ontology with better reliability is selected as the source ontology. The reliability of an ontology is measured by the semantic accuracy, which is computed through the average of the squared semantic distance between each concept  $c_i$  and the ontology  $O$ 's taxonomic root node  $ROOT$  [29]. In particular, the formula of calculating semantic accuracy is presented as follows:

$$semAccuracy(O) = \frac{\sum_{c_i \in C} semDistance(c_i, ROOT)^2}{|C|} \quad (7)$$

where  $semDistance(c_i, ROOT) = \log_2(1 + \frac{|Ances(c_i)|-1}{|Ances(c_i)|})$  calculates the semantic distance between the concept  $c_i$  and  $ROOT$ .  $Ances(c_i)$  refers to the set of taxonomic ancestors of concept  $c_i$  in the ontology including itself.

The source ontology is partitioned into disjoint segments through an ontology partition algorithm which is extended from SCAN [30]. Then, a concept relevance measure based approach is adopted to determine the similar target ontology segments of each source ontology segment  $seg_{src}$ . Particularly, for each target ontology concept  $c_i$ , the similarity value  $sim_{c_i}$  between  $c_i$  and  $seg_{src}$  is calculated by summing up every  $SMOA(c_i, c_j)$  (see also Section 7.1). If  $sim_{c_i}$  is larger than the threshold,  $c_i$  will be added to candidate concept set  $C_{candidate}$ . If the relevance value of a concept in  $C_{candidate}$  is bigger than the threshold, it will be added to the final target segment. Given a concept  $c_m \in C_{candidate}$ , the relevance value of  $c_m$  to source ontology segment can be calculated by the following formula:

$$relevance(c_m) = sim_{c_m} \times \sum_{c_n \in C_{candidate}} sim_{c_n} \times e^{-(sPath(c_m, c_n))^2} \quad (8)$$

where  $sim_{c_m}$  and  $sim_{c_n}$  respectively denote the similarity value of  $c_m$  and  $c_n$  to  $seg_{src}$ ,  $sPath(c_m, c_n)$  is the shortest length between their corresponding vertexes in ontology taxonomy structure.

After partitioning the ontologies, the matching process only needs to deal with the similar ontology segments' matching problem, and all the similarity values obtained in the process of ontology partitioning are stored in hash map to avoid repeating calculations in the hereafter matching process. With respect to the details of the alignment-oriented ontology partition algorithm, please see also [29]. Finally, for each segment alignment, the mappings with SMOA value lower than *upperThreshold* are filtered, and the rest are presented to user for validation. The mappings that are selected as correct by the user are utilized to initialize the PRA. Since users cannot validate too many mappings at a time, *upperThreshold* should be high enough to ensure the reasonable volume of suggested mappings and their correctness to the maximum extent. However, if the value is too large, many truly correct mappings might be filtered out. Therefore, the suggested range of *upperThreshold* is [0.7, 0.9]. Through the preliminary experiment, it is found that *upperThreshold* = 0.8 works better.

## 6.2. Candidate Mapping Determination and Conflict Mapping Validation

When ICHEA gets stuck, i.e., the elite keeps unchanged for  $\epsilon$  generations, the user needs to guide the search direction of the algorithm by removing the wrong correspondences, and adding mappings that are not detected, as well as updating the PRA. The details are as follows:

- ask the user to validate the correspondences in  $EVSet = \{EV_i \in EM | E_i^{elite} = -1 \wedge X_i^{globalBest} = -1\}$ . Set  $P_i^{elite} - \theta/2, P_i^{localBest} - \theta/2, P_i^{globalBest} + \theta$ . If the user judges  $X_i^{localBest}$ 's corresponding mapping as correct, set  $X_i^{globalBest} = X_i^{localBest}$ ;
- ask the user to validate the correspondences in  $EVSet = \{EV_i, EV_j \in EM | X_i^{globalBest} = -1 \wedge X_j^{globalBest} = -1 \wedge X_i^{elite} = X_j^{elite}\}$ . If the user judges  $X_i^{elite}$  as correct, set  $X_i^{globalBest} = X_i^{elite}, P_i^{elite} - \theta/2, P_i^{localBest} - \theta/2, P_i^{globalBest} + \theta$ . For the rest EVs in  $EVSet$ , set  $X^{elite}$  with corresponding similarity value lower than *lowerThreshold* as -1, and  $P^{elite} - \theta/2, P^{localBest} - \theta/2, P^{globalBest} + \theta$ . Otherwise, if none is correct, for each EV in  $EVSet$ , set  $X^{elite}$  with corresponding similarity value lower than *lowerThreshold* as -1, and  $P^{elite} - \theta/2, P^{localBest} - \theta/2, P^{globalBest} + \theta$ ;

In the first step, the user is required to add new correspondences in terms of improving solution's completeness. In the second step, negative correspondences are removed to increase the soundness of the solution. By presenting

the problematic correspondences to the user in the first step, this approach can reduce the number of candidate correspondences while helping ICHEA search for better solutions. In addition, it is unnecessary for a user to validate all the conflicting correspondences, because given the golden alignment is one to one, if the user validates one correspondence as true, the other correspondences related to this concept in this correspondence will be rejected. Therefore, in the second step, this method presents several conflicting mappings simultaneously to the user. When the user selects one mapping as correct, the rest will be automatically rejected, which further reduces user workload and improves the validating efficiency.

## 7. Increase User Involvement's Value

Propagating user validation is an effective approach to increase the value of user involvement. However, the validation process is a difficult cognitive task, which requires tremendous patience and expert understanding of the ontology domain, terminology, and semantics [31]. On one hand, the user is generally expected to make fewer errors than automated systems. On the other hand, there are risks to take user's inputs for granted, since the user could make mistakes during the validating process. Therefore, in this work, only the correspondences which are judged to be positive by both the similarity measure and the user will be used for the propagation. In the following, the asymmetrical profile-based similarity measure is presented for measuring the ontology entity similarity, and the propagating approach is described, which can increase the value of user validation under various user error rate cases.

### 7.1. *Asymmetrical Profile-based Similarity Measure*

The foundation of ontology matching technique is the similarity measure of ontology entities [32], which is the function that calculates the degree to which entities are similar to one another. In this work, an asymmetrical profile-based similarity measure is proposed to calculate the entities' similarity values. To be specific, firstly, a profile for each ontology entity is constructed by collecting the label, comment, and property labels from itself, and all its direct descendants. Then, given two entities  $e_1$  and  $e_2$  and their corresponding profiles  $p_1$  and  $p_2$ , the similarity value of  $p_1$  and  $p_2$  can

be calculated by the following two asymmetrical measures:

$$sim_1(p_1, p_2) = \frac{|p_1 \cap p_2|}{|p_1|} \quad (9)$$

$$sim_2(p_1, p_2) = \frac{|p_1 \cap p_2|}{|p_2|} \quad (10)$$

where  $|p_1|$  and  $|p_2|$  are the cardinality of the profile  $p_1$  and  $p_2$  respectively,  $|p_1 \cap p_2|$  is the number of identical elements in  $p_1$  and  $p_2$ .

On this basis, the similarity of  $e_1$  and  $e_2$  is calculated through the following formula:

$$sim(e_1, e_2) = \begin{cases} \frac{sim_1(p_1, p_2) + sim_2(p_1, p_2)}{2}, & \text{if } |sim_1(p_1, p_2) - sim_2(p_1, p_2)| \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

In this study,  $\delta$  is the threshold to measure the extent of the semantic equivalence between  $sim_1(p_1, p_2)$  and  $sim_2(p_1, p_2)$ . When the similarity value between two profile elements is above the threshold, they are identified as semantically similar. Generally,  $\delta$  should be set relatively small to reflect that there are little differences between  $sim_1(e_1, e_2)$  and  $sim_2(e_1, e_2)$ , i.e.,  $e_1$  and  $e_2$  are semantically equivalent. However, if  $\delta$  is too small, many semantically equivalent terms will be lost. Therefore, the suggested range of  $\delta$  is  $[0.01, 0.15]$ . In this work, to obtain a suitable  $\delta$ , a pre-experiment is conducted on the bench-mark by varying the value of  $\delta$  in its suggested range, and found the semantic equivalence performed better when  $\delta = 0.1$ .

The similarity value of two profile elements is calculated by SMOA [33], which is the most performing syntax measure for the ontology matching problem, and a linguistic measure, which calculates a synonymy-based distance through an electronic lexical database WordNet [34]. Given two words  $w_1$  and  $w_2$ , their similarity  $sim(w_1, w_2)$  is calculated according to the following formula:

$$sim(w_1, w_2) = \begin{cases} 1, & \text{if two words are synonymous} \\ SMOA(w_1, w_2), & \text{otherwise} \end{cases} \quad (12)$$

Particularly, SMOA distance between two strings  $s_1$  and  $s_2$  can be defined as  $SMOA(s_1, s_2) = Comm(s_1, s_2) - Diff(s_1, s_2) + WinklerImpr(s_1, s_2)$ , where  $Comm(s_1, s_2)$  stands for the commonality between  $s_1$  and  $s_2$ ,  $Diff(s_1, s_2)$  for the difference, and  $WinklerImpr(s_1, s_2)$  for the improvement of the result

from using the method introduced by Winkler [35]. During the automatic matching process, the asymmetrical profile-based similarity measure is used to measure the correspondence's confidence. During the user validating process, only the correspondence that is judged as correct by the user and also has high similarity value could propagate its confidence to the neighborhood, which mitigates the negative impacts brought by the human error.

### 7.2. User Validation Propagation

When the scale of the ontology is large, validating the selected correspondences only is far from sufficient. Therefore, the value of user involvement must be increased by propagating the user validations to determine other potential matchings between two ontologies. Based on the intuition that elements of two distinct ontologies will be similar when their adjacent elements are similar, and the observation that a correct alignment should not be inconsistent with ontology's concept hierarchies, which is organized by the "is-a" properties [36], ICHEA propagates each value that has been validated by the user and is also of high similarity to its source entity's ancestors' correspondences. For each user validated mapping with a similarity value bigger than  $upperThreshold$ , e.g.,  $X_i^{globalBest}$ , ICHEA will automatically check all the  $i$ -th source entity's ancestors. Assuming the  $j$ -th source entity is one of its ancestor, if  $X_j^{globalBest}$  is -1 and the  $k$ -th target entity is the ancestor of  $X_i^{globalBest}$ -th target entity and their similarity value should be larger than  $upperThreshold$ , the value of  $X_j^{globalBest}$  will then be updated as  $k$  and  $P_j^{elite} - st/2$ ,  $P_j^{localBest} - st/2$  and  $P_i^{globalBest} + st$ ; when  $X_j^{globalBest}$ -th target entity is the ancestor of  $X_i^{globalBest}$ -th target entity,  $P_j^{elite} - st/2$ ,  $P_j^{localBest} - st/2$  and  $P_i^{globalBest} + st$ ; when  $X_j^{globalBest}$ -th target entity is not the ancestor of  $X_i^{globalBest}$ -th target entity,  $P_j^{elite} + st/2$ ,  $P_j^{localBest} + st/2$  and  $P_i^{globalBest} - st$ .

Since validation process is a difficult cognitive task, it is risky to take the user's inputs for granted, which may lead to the propagation of errors. To mitigate the negative effects brought by propagating wrong user validations, in this approach, only those correspondences with high similarity values can be used to propagate their confidence. In addition, the matching probability is utilized instead of similarity value to determine the ontology alignment, i.e. our propagation algorithm does not alter the similarity value but the correspondence's confidence of holding true. Thus, under EA's survival of the fittest, ICHEA can effectively decrease the negative effects brought by



erroneous user validation and improve the quality of the alignment under 0.3 error rate case.

## 8. Outline of Partial Reference Alignment based Interactive Compact Hybrid Evolutionary Algorithm

The working flow of ICHEA is given in Figure 4. First, all the control parameters of ICHEA are set up, please see also Section 5.1. After that, we randomly selected the number in  $\{0, 1, \dots, |seg_{tgt}|\}$  to initialize EM's element  $X_i^{elite}$  and  $X_i^{localBest}$ , and set  $X_i^{globalBest} = -1, i = 1, 2, \dots, |seg_{src}|$ . Then, we initialized PM by setting  $P_i^{elite} = 0.25, P_i^{localBest} = 0.25, X_i^{globalBest} = 0.5, i = 1, 2, \dots, |seg_{src}|$ , which make the algorithm focus on the exploration at the beginning. At last, we generate an individual by EM and PM to initialize the elite. During each generation of ICHEA, a new individual is generated through EM and PM to compare with the current elite in terms of their fitness values. EM, PM and elite are updated based on the better solution, with the aim to move the EM and PM toward the better solution. Then, the local search process on elite is executed to update EM, PM and elite. If ICHEA got stuck, i.e., the elite kept unchanged for  $\epsilon$  generations, the user validation process will be activated to update the EM and PM, and then the algorithm went to the next generation. Otherwise, the algorithm will directly go to the next generation. When the termination condition is met, the algorithm stopped. In this work, all the segment matching processes are executed in parallel. After segment alignments are obtained, they are aggregated into a final ontology alignment through a greedy strategy. Among the corresponding correspondences of each source ontology entity, we selected the one with the highest confidence value to form the final ontology alignment. Finally, the quality of final ontology alignment is evaluated with f-measure.

ICHEA represents the population as a probability distribution over the solution set, and simulates the behavior of the traditional EA with uniform crossover. In this way, it processes each gene independently, and requires less memory than the traditional EA does. The local refinement strategy is introduced into the traditional EA's evolutionary process, which allows to increase the convergence speed, and reduce the runtime needed. Moreover, the introduced user validation process leverages user's insights to the solution, to guide the algorithm's search, and improve the search ability of EA and the solution's quality.

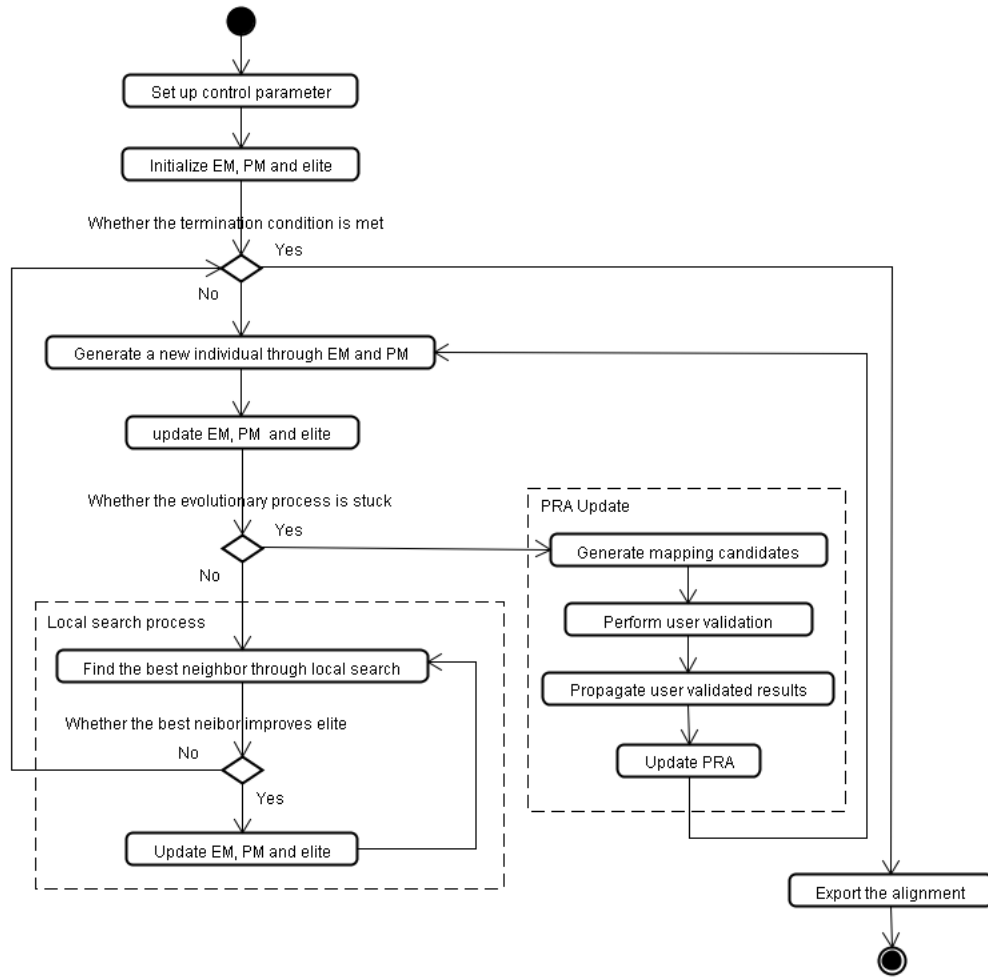


Figure 4: Outline of the PRA-based Interactive Compact Hybrid Evolutionary Algorithm

## 9. Experimental Studies and Analysis

In this experiment, the well-known OAEL 2016 benchmark track and interactive track are exploited to test the performance of ICHEA. Benchmark track <sup>2</sup> consisted of 50 test cases. Each test case has a bibliographic seed ontology, a variant of the seed ontology and a reference alignment. In this experiment, we use the downloadable datasets from the OAEI 2016 official web

<sup>2</sup><http://oaei.ontologymatching.org/tests>

Table 1: The Overview of the Parameter Settings in ICHEA

| Symbol     | Meaning  | Value   |
|------------|--|---|
| $\epsilon$ | the upper threshold that elite keeps unchanged | 20  |
| $\theta$   | user's trust extent                            | 0.5   |
| $st$       | step size                                      | 0.01  |
| $C$        | elite's neighborhood scale                     | $5 + \lceil \max\{ Seg_{src} ,  Seg_{tgt} \} / 10 \rceil$ |
| $gen$      | maximum generation                             | 300   |

site for testing purpose. Interactive track <sup>3</sup> consists of three datasets in the OAEI 2016, i.e., Anatomy <sup>4</sup>, Conference <sup>5</sup> and Large Biomedical Ontologies (Large Bio) <sup>6</sup>. In this track, an oracle is used to simulate the domain expert's behaviors, which mitigates the negative effect brought by different levels of expertise and human biases, and minimizes the user's consumption of time in each validating process. As a result, the effectiveness of each system in exploiting user validation can be better evaluated. The Anatomy dataset consists of two ontologies, i.e., the Adult Mouse Anatomy (AMA) ontology and a part of the National Cancer Institute Thesaurus (NCI) that describes the human anatomy. The Conference dataset covers 16 ontologies that describe the domain of conference organization. We only use the test case where the reference alignment is available. Large Bio consists of 6 tasks in different sizes ranging from tens to hundreds of even thousands of classes, aiming to find out alignments among Foundational Model of Anatomy (FMA), SNOMED CT, and National Cancer Institute Thesaurus (NCI).

### 9.1. Experiment Setup

The configuration of ICHEA is given in Table 1.

In addition, in order to compare with the participants of OAEI, Conference and Anatomy tracks are run on a server with 3.46 GHz (6 cores) with 8GB RAM. In the meantime, Large Bio track is run with an Intel Core i7-4600U CPU @ 2.10GHz x 4 with 15GB RAM being allocated. The experiment allows ICHEA to request an oracle, i.e., ICHEA presents a cor-

<sup>3</sup><http://oaei.ontologymatching.org/2016/interactive/index.html>

<sup>4</sup><http://oaei.ontologymatching.org/2016/anatomy/index.html>

<sup>5</sup><http://oaei.ontologymatching.org/2016/conference/index.html>

<sup>6</sup><http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/>

respondence to the oracle, which then tells the algorithm whether the correspondence is right or wrong. Besides, the oracle with variable error rates is considered to reflect a more realistic scenario where a user can not always provide correct answers. Three different error rates are tested in the experiment, which are 0.1, 0.2 and 0.3 respectively. These error rates are randomly introduced into the reference alignment with given rates. The results of ICHEA are the average of thirty independent runs.

The configuration of ICHEA in our work follows below principles:

- $\epsilon$  is the threshold that ICHEA used to determine the timing of getting user involved. If  $\epsilon$  is too big, the matching process would become a solely automatic matching process, and the obtained results would be closed to CHEA-based approach. If  $\epsilon$  is too small, the user needs to invalidate the alignment frequently, which would greatly increase their workload. Moreover, the erroneous user validation will be difficult to be removed by the automatic evolving process. Therefore, the suggested range of  $\epsilon$  is [15, 35]. In the preliminary experiment, we found that the results obtained with  $\epsilon = 20$  are acceptable for various heterogeneous problems in all testing datasets.
- $\theta$  is utilized to update PM based on user validation. If  $\theta$  is close to 1, users' erroneous validations would be difficult to be removed. Moreover, with the rise of user error rate, the quality of obtained alignments will drop sharply. If  $\theta$  is close to 0, users' validations would have a tiny effect on the search direction of algorithm, and the obtained results would be close to CHEA-based approach. Therefore, the suggested range of  $\theta$  is [40, 60]. In the preliminary experiment, we found that  $\theta = 0.5$  worked better.
- $st$  is utilized to update PM based on the comparison between the elite and newly generated solution. If  $st$  is too big, the probabilities of all mappings would quickly become 1 or 0, which probably leads to premature converge. If  $st$  is too small, ICHEA would become a stochastic algorithm. Therefore, the suggested range of crossover probability is [0.05, 0.15]. In the preliminary experiment, we found that the results obtained with  $st = 0.1$  are the best in all testing datasets.
- The elite's neighborhood scale and maximum generation for termination depend on the scale of the problem, the suggested ranges for them

are  $\lceil \max\{|Seg_{src}|, |Seg_{tgt}|\}/10 \rceil$ ,  $10 + \lceil \max\{|Seg_{src}|, |Seg_{tgt}|\}/10 \rceil$  and  $[200, 500]$ , respectively. Since the problem scale is not that severe when matching segment pair, we set the elite's neighborhood scale and maximum generation as  $C = 5 + \lceil \max\{|Seg_{src}|, |Seg_{tgt}|\}/10 \rceil$  and 300, respectively.

Table 3 shows the mean values of all testing cases in benchmark track, which are respectively obtained by OAEI's participants, three state-of-the-art EA-based ontology matching techniques, and ICHEA. Tables 4, 6 and 7 demonstrate the results obtained by OAEI's participants and ICHEA on interactive Anatomy, Conference and Large Bio track, respectively. The symbols  $r$ ,  $p$  and  $f$  in these tables stand for recall, precision and f-measure, respectively. F-measure per second in Table 3 is calculated by dividing f-measure by runtime, which is used by OAEI 2016 to measure the efficiency of the ontology matcher [37]. The symbols  $\bar{f}$ ,  $\bar{r}$  and  $\bar{p}$  represent the non-interactive version of those ontology matching technique's f-measure, recall and precision. Since the effectiveness with which systems exploit the user validation can not only be evaluated by f-measure, recall and precision, but also the number of requests asked [3], the total requests and the runtime of each ontology matching methods are presented. Finally, Table 9 and Table 8 give the statistical comparisons among ICHEA and other ontology matchers.

## 9.2. Experimental Result and Analysis

### 9.2.1. Benchmark Track

The benchmark track consists of a set of small scale ontologies which are built around a seed ontology, which contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals, and its variations, which are artificially generated, and focus on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups: simple tests (1XX) compares the reference ontology with itself; systematic tests (2XX) are obtained by discarding/modifying features, which include names of entities, comments, the specialization hierarchy, instances, properties and classes, from the reference ontology; real-life ontologies (3XX) are found on the web. Table 2 illustrates a brief description of OAEI 2016's benchmark track.

A preliminary test is firstly carried out to compare the performance over time among three state-of-the-art EA-based ontology matching techniques,

Table 2: Brief description on benchmark track. 1XX, 2XX and 3XX stands for the test case whose ID beginning with the prefix digit 1, 2 and 3, respectively

| ID  | Description  |
|-----|--|
| 1XX | The ontologies under alignment are the same or the first one is the OWL Lite restriction of the second one |
| 2XX | The ontologies under alignment have different lexical, linguistic or structure features                    |
| 3XX | The ontologies under alignment are real world cases  |

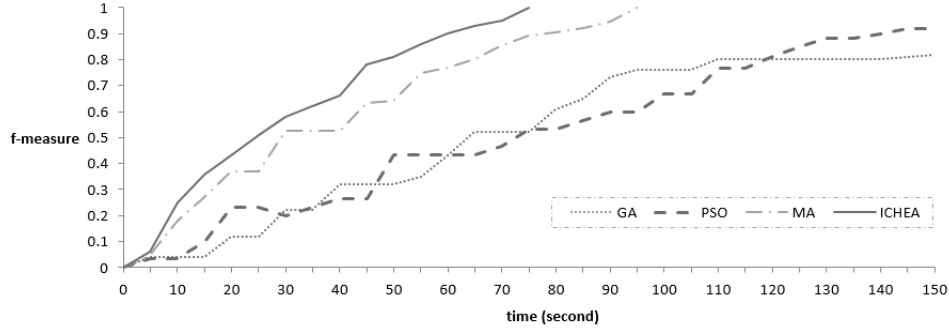


Figure 5: The performance over time among three state-of-the-art EA-based ontology matching techniques and ICHEA

i.e., GA-based [38], MA-based [7], and PSO-based [39] ontology matching approaches. The results are shown in Figure 5. The graphics plot on  $y$  axis are the f-measure values computed for the alignments provided by different algorithms, and the  $x$  axis represents the run time. Figures depict that our proposal is able to determine a high-quality alignment in a short time.

In Table 3, the automatic ontology matching techniques used by OAEI 2016's participants is employed to compare with ICHEA in terms of the quality of alignments. In particular, AgreementMakerLight (AML) [15] is an automated ontology matching system primarily depending on element-level matching and the use of external resources as background knowledge. Its matching workflow consists of three sequential steps: applying different matching processes (such as background knowledge matching, structural matching and property matching), greedy selection algorithm based mapping selection, and heuristic repair algorithm based alignment repair to ensure that the final alignment is coherent. CroMatcher [40] uses the weighted aggregation method to automatically determine the aggregating weight of several basic matchers, and the iterative final alignment method to determine the final ontology alignment by selecting appropriate correspondences

from the aggregated matching results. Lily [41] first constructs a semantic subgraph for each ontology entity for the similarity computation. Then, the text matching and structure matching techniques are combined to determine the most similar entities. Finally, a mapping debugging technique is utilized to detect and correct the mapping errors to improve the quality of the final alignment. LogMap family [16] consists of three matching systems, i.e., LogMap, LogMapLt and LogMap-Bio. LogMapLt is a lightweight variant of LogMap, which only applies string matching techniques. LogMapBio includes an extension to use a mediating biological ontology as the external resource. LogMap first utilizes the lexical information of each entity to initialize a mapping set of manageable size. Afterwards, it utilizes sophisticated reasoning (such as logic-based extraction, propositional reasoning and axiom tracking) and repairs techniques to maximize the ontology alignment's logical consistency: (1) the mappings should not lead to unsatisfying classes in the integrated ontology, and (2) the mappings should link entities that have similar neighborhood. XMap [17] uses three different layers to perform the ontology matching process, i.e., a terminological layer, a structural layer and an alignment layer. In particular, the terminological layer utilizes the similarity measure based on UMLS [42] and WordNet to determine the similarity value between the entity names. The structural layer computes the similarity between the concepts by taking into account the taxonomic hierarchy, and calculates the similarity using the information of the internal structure of concepts, i.e., their properties, types and cardinality restrictions. The alignment layer aims to provide the final similarity matrix between the concepts taking into account the influence of the number of properties and the value of similarity that properties bring to the final similarity between them. The output values of each layer serve as the input to the upper one. Each layer provides an improvement to the computation of the similarity between concepts. Pheno family [43] consists of three matching systems, i.e., PhenoMM, PhenoMF and PhenoMP, which rely on different versions of the PhenomeNET [44] ontology with variable complexity. They first use PhenomeNET ontology based reasoning technique to identify the alignment between ontologies, and then generate the additional mappings through AML.

We can see that CroMatcher and Lily are two top matching systems among OEAI participants with 0.89 f-measure, whose precisions are very high (0.96 and 0.97 respectively). AML, XMap and LogMap family can not return with very good alignments. The Pheno family reverted with huge but poor alignments. Concerning three EA-based ontology matching techniques,

Table 3: Comparison between OAEI 2016's participants, three state-of-the-art EA-based ontology matching techniques and ICHEA on benchmark track. The numbers are the mean values of all testing cases, and  $r$ ,  $p$  and  $f$  in the table stand for recall, precision and f-measure, respectively.

| Matching System | $r$  | $p$  | $f$  | Runtime (sec) | f-measure per second |
|-----------------|------|------|------|---------------|----------------------|
| AML             | 0.24 | 1.00 | 0.38 | 120           | $3.17 \times e^{-3}$ |
| CroMatch        | 0.83 | 0.96 | 0.89 | 1100          | $8.09 \times e^{-4}$ |
| Lily            | 0.83 | 0.97 | 0.89 | 2211          | $4.03 \times e^{-4}$ |
| LogMap          | 0.39 | 0.93 | 0.55 | 194           | $2.84 \times e^{-3}$ |
| LogMapLt        | 0.50 | 0.43 | 0.46 | 96            | $4.79 \times e^{-3}$ |
| PhenoMF         | 0.01 | 0.03 | 0.01 | 1632          | $6.13 \times e^{-6}$ |
| PhenoMM         | 0.01 | 0.03 | 0.01 | 1743          | $5.74 \times e^{-6}$ |
| PhenoMP         | 0.01 | 0.02 | 0.01 | 1833          | $5.31 \times e^{-6}$ |
| XMap            | 0.40 | 0.95 | 0.56 | 123           | $4.55 \times e^{-3}$ |
| LogMapBio       | 0.24 | 0.48 | 0.32 | 54439         | $5.88 \times e^{-6}$ |
| GA              | 0.76 | 0.68 | 0.71 | 240           | $2.99 \times e^{-3}$ |
| PSO             | 0.66 | 0.72 | 0.68 | 224           | $3.07 \times e^{-3}$ |
| MA              | 0.78 | 0.81 | 0.79 | 182           | $4.36 \times e^{-3}$ |
| ICHEA           | 0.82 | 0.98 | 0.90 | 152           | $5.99 \times e^{-3}$ |



Table 4: Comparison between OAEI 2016’s participants and ICHEA on interactive anatomy track. The symbols  $r$ ,  $p$  and  $f$  in the table stand for recall, precision and f-measure, respectively, and  $\bar{f}$ ,  $\bar{r}$  and  $\bar{p}$  respectively stand for the system’s non-interactive version’s f-measure, recall and precision.

| Error Rate |                              | ALin              | AML               | LogMap            | XMap              | ICHEA             |
|------------|------------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 0.0        | $\bar{f}(\bar{r}, \bar{p})$  | 0.50 (0.33, 0.98) | 0.94 (0.93, 0.95) | 0.87 (0.84, 0.91) | 0.89 (0.86, 0.92) | 0.82 (0.80, 0.84) |
|            | $f(r, p)$                    | 0.85 (0.74, 0.99) | 0.95 (0.94, 0.96) | 0.90 (0.84, 0.98) | 0.89 (0.86, 0.92) | 0.95 (0.94, 0.95) |
|            | Total Requests               | 803               | 241               | 590               | 35                | 262               |
|            | Mean Improvement per Request | 0.04%             | 0.004%            | 0.005%            | 0.00%             | 0.05%             |
|            | Runtime (sec)                | 505               | 48                | 27                | 49                | 25                |
| 0.1        | $f(r, p)$                    | 0.80 (0.70, 0.94) | 0.94 (0.94, 0.95) | 0.89 (0.83, 0.96) | 0.89 (0.86, 0.92) | 0.94 (0.93, 0.94) |
|            | Total Requests               | 769               | 273               | 612               | 35                | 280               |
|            | Mean Improvement per Request | 0.04%             | 0.00%             | 0.003%            | 0.00%             | 0.05%             |
|            | Runtime (sec)                | 489               | 50                | 24                | 46                | 25                |
| 0.2        | $f(r, p)$                    | 0.76 (0.66, 0.90) | 0.94 (0.94, 0.93) | 0.87 (0.81, 0.94) | 0.89 (0.86, 0.92) | 0.94 (0.93, 0.94) |
|            | Total Requests               | 750               | 300               | 645               | 35                | 264               |
|            | Mean Improvement per Request | 0.03%             | 0.00%             | 0.00%             | 0.00%             | 0.05%             |
|            | Runtime (sec)                | 481               | 48                | 24                | 47                | 24                |
| 0.3        | $f(r, p)$                    | 0.72 (0.61, 0.86) | 0.93 (0.92, 0.95) | 0.87 (0.81, 0.93) | 0.89 (0.86, 0.92) | 0.93 (0.93, 0.92) |
|            | Total Requests               | 740               | 308               | 650               | 35                | 257               |
|            | Mean Improvement per Request | 0.03%             | -0.03%            | 0.00%             | 0.00%             | 0.04%             |
|            | Runtime (sec)                | 472               | 47                | 24                | 47                | 26                |

because of the introduction of the local search strategy, the results clarify that MA outperforms GA and PSO. In this track, the quality of ICHEA is higher than what all of OAEI 2016’s participants do, as well as that of EA-based techniques, in terms of f-measure and f-measure per second. In particular, the precision of ICHEA is the highest, which shows the effectiveness of the asymmetrical profile-based similarity measure and the exploitation of user validation. Therefore, ICHEA is able to efficiently determine the high-quality small-scale ontology alignment.

### 9.2.2. Interactive Anatomy Track

The anatomy track is a large ontology matching task about matching the Adult mouse anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes) which describes the human anatomy. Adult mouse anatomy is a structured controlled vocabulary describing the anatomical structure of the adult mouse, whereas NCI depicts the human anatomy for the purpose of cancer research.

Table 4 presents the results for the anatomy dataset under four different error rate cases. When systems are evaluated with an all-known oracle (i.e., 0.0 error rate case), ICHEA obtains the highest f-measure. In particular, comparing with CHEA, ICHEA shows balanced improvements on both recall and precision, by 17.50% and 13.09%. This means that in ICHEA, user validation is beneficial, by introducing the new non-trivial mappings and filtering the trivial ones. In addition, 0.05% mean improvement per request

in ICHEA is also higher than that of other systems, with the runtime of 25 seconds being the lowest. In summary, ICHEA makes effective exploitation of the oracle to achieve the great improvement in terms of efficiency.

When introducing the error to the oracle's answers, each ontology matching technique's performance starts to decrease. However, in the 0.1, 0.2 and 0.3 cases respectively, ICHEA only loses 1%, 1% and 2% in f-measure. With respect to XMap, since its requests are mainly about the judgment of incorrect mappings in the post-process phase, its performance does not change at all with the increase of the error rate. However, ICHEA requires the user to judge both correct and incorrect mappings, in the meantime, adding new mappings in the matching process, meaning it relies more on the user validation. In addition, the mean improvement per request of ICHEA is higher than that of the other matching systems, which also shows that it can better make use of the user's interactions. Comparing with CHEA's results, the error rate of 0.3 in user validation still proves the benefits of ICHEA in terms of 13.41% improvement on f-measure. Moreover, the average runtime needed by ICHEA is also lower than that needed by most of OAEI participants. In conclusion, ICHEA is able to efficiently utilize the user interaction to achieve the best quality of alignment.

### 9.2.3. Interactive Conference Track

The goal of this track is to find alignments within a collection of ontologies describing the domain of organising conferences, which contains 16 ontologies in the domain of conference organization and the total number of testing cases is 120. In addition, these ontologies differ in their numbers of classes and properties, in expressivity, but also in underlying resources. Table 5 shows a brief description of OAEI 2016's conference track.

Table 6 demonstrates the average results on interactive conference track by OAEI 2016's participants and ICHEA. It can be seen that when systems are evaluated with an all-known oracle, ICHEA outperforms the other systems in terms of f-measure. Meanwhile, the improvement on CHEA is approximately 62%, which is much higher than that of other OAEI participants. In addition, the recall and precision improvements are also outstanding, with a percentage of 68% and 57% respectively. The comparison also shows that the substantial improvement of ICHEA is more supported by gains in precision. With respect to improvement per request, its output is 0.16%, which is also higher than that of the other matching systems. In addition, it takes only 21 seconds for ICHEA to determine the alignment, which ranks first

Table 5: Brief description on conference track.

| Ontology's Name | Number of Classes | Number of Datatype Properties | Number of Object Properties |
|-----------------|-------------------|-------------------------------|-----------------------------|
| Ekaw            | 74                | 0                             | 33                          |
| Sofsem          | 60                | 18                            | 46                          |
| Sigkdd          | 49                | 11                            | 17                          |
| Iasted          | 140               | 3                             | 38                          |
| Micro           | 32                | 9                             | 17                          |
| Confious        | 57                | 5                             | 52                          |
| Pcs Tool        | 23                | 14                            | 24                          |
| OpenConf        | 62                | 21                            | 24                          |
| ConfTool        | 38                | 23                            | 13                          |
| Crs Tool        | 14                | 2                             | 15                          |
| Cmt Tool        | 36                | 10                            | 49                          |
| Cocus           | 55                | 0                             | 35                          |
| Paperdyne       | 47                | 21                            | 61                          |
| Edas            | 104               | 20                            | 30                          |
| MyReview        | 39                | 17                            | 49                          |
| Linklings       | 37                | 16                            | 31                          |

Table 6: Comparison between OAEI 2016's participants and ICHEA on interactive conference track. The symbols  $r$ ,  $p$  and  $f$  in the table stand for recall, precision and f-measure, respectively, and  $\bar{f}$ ,  $\bar{r}$  and  $\bar{p}$  respectively stand for the system's non-interactive version's f-measure, recall and precision.

| Error Rate |                              | ALin              | AML               | LogMap            | XMap              | ICHEA             |
|------------|------------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 0.0        | $\bar{f}(\bar{r}, \bar{p})$  | 0.40 (0.25, 0.88) | 0.73 (0.65, 0.84) | 0.68 (0.59, 0.81) | 0.68 (0.57, 0.83) | 0.52 (0.50, 0.54) |
|            | $f(r, p)$                    | 0.83 (0.73, 0.95) | 0.79 (0.71, 0.91) | 0.72 (0.61, 0.88) | 0.68 (0.57, 0.83) | 0.84 (0.84, 0.85) |
|            | Total Requests               | 326               | 271               | 142               | 4                 | 194               |
|            | Mean Improvement per Request | 0.10%             | 0.02%             | 0.03%             | 0.00%             | 0.16%             |
|            | Runtime (sec)                | 101               | 29                | 26                | 21                | 20                |
| 0.1        | $f(r, p)$                    | 0.72 (0.67, 0.79) | 0.76 (0.70, 0.84) | 0.70 (0.60, 0.84) | 0.68 (0.57, 0.83) | 0.80 (0.79, 0.82) |
|            | Total Requests               | 315               | 285               | 140               | 4                 | 192               |
|            | Mean Improvement per Request | 0.10%             | 0.01%             | 0.01%             | 0.00%             | 0.14%             |
|            | Runtime (sec)                | 101               | 30                | 26                | 22                | 20                |
| 0.2        | $f(r, p)$                    | 0.64 (0.61, 0.67) | 0.72 (0.68, 0.76) | 0.68 (0.58, 0.82) | 0.68 (0.57, 0.83) | 0.74 (0.71, 0.76) |
|            | Total Requests               | 303               | 290               | 143               | 4                 | 212               |
|            | Mean Improvement per Request | 0.08%             | -0.003%           | 0.00%             | 0.00%             | 0.10%             |
|            | Runtime (sec)                | 100               | 33                | 26                | 21                | 24                |
| 0.3        | $f(r, p)$                    | 0.56 (0.56, 0.57) | 0.68 (0.65, 0.71) | 0.67 (0.58, 0.80) | 0.68 (0.57, 0.83) | 0.71 (0.68, 0.75) |
|            | Total Requests               | 303               | 284               | 144               | 4                 | 224               |
|            | Mean Improvement per Request | 0.05%             | -0.02%            | -0.007%           | 0.00%             | 0.08%             |
|            | Runtime (sec)                | 99                | 30                | 26                | 22                | 25                |

among all the matching systems. From above, ICHEA requires only a small number of requests to the oracle to achieve the great improvement on the efficiency.

When error rates are introduced, with the error rates of 0.1, 0.2 and 0.3, the loss in f-measure from ICHEA are 4.76%, 11.9% and 15.47% respectively, which are better than those of OAEI's participants and CHEA. In particular, under the error rate of 0.3, both recall and precision of ICHEA are higher than CHEA's results, which means that even when the user error rate is 0.3, ICHEA can still improve the quality through user validation. Due to the utilization of analyzing multiple conflicting mappings, ICHEA is proved to be helpful in achieving higher mean improvements per request than other systems do. Moreover, the average runtime for ICHEA under all three cases is the lowest among all the other OAEI participants. Therefore, compared to state-of-the-art ontology matching systems, ICHEA determines the highest quality of alignment under different user error rate cases, with lower runtime and a relatively smaller number of requests.

#### 9.2.4. *Interactive Large Bio Track*

Large Bio track aims to find alignments between the large and semantically rich biomedical ontologies FMA, SNOMED CT, and NCI, which contains 78,989, 306,591 and 66,724 classes, respectively. The track has been split into three matching problems: FMA-NCI, FMA-SNOMED and SNOMED-NCI, and each matching problem in three tasks involving different fragments of the input ontologies.

As is depicted in Table 7, with an all-knowing oracle, ICHEA improves the performance of the non-interactive version by 15.78% in terms of f-measure, while that of LogMap is 3.2%, AML is 2.6%, and ServOMBI is 1.4%. Additionally, while AML shows its improvement in terms of recall, LogMap has the strength in precision, and ServOMBI improved essentially only on precision, the improvement of ICHEA is supported by the gains from both recall and precision. In conclusion, ICHEA's alignment quality is better than all other systems and CHEA, while requesting the shortest runtime.

The introduction of user errors poses different influences on different systems. In ICHEA, the average performance dropped by 4.92% under the three cases, AML 3%, and LogMap 5.70%, while ServOMBI decreases by 16.6%. On top of that, the f-measure of ICHEA is higher than those of all other OAEI participants and CHEA under each circumstance. With the increase of user errors, the number of requests from ICHEA to the oracle increased

Table 7: Comparison between OAEI 2015’s participants and ICHEA on interactive large bio track. The symbols  $r$ ,  $p$  and  $f$  in the table stand for recall, precision and f-measure, respectively, and  $\bar{f}$ ,  $\bar{r}$  and  $\bar{p}$  respectively stand for the system’s non-interactive version’s f-measure, recall and precision.

| Error Rate |                              | AML                  | LogMap                | ServOMBI              | ICHEA                |
|------------|------------------------------|----------------------|-----------------------|-----------------------|----------------------|
| 0.0        | $\bar{f}(\bar{r}, \bar{p})$  | 0.81 (0.75, 0.90)    | 0.79 (0.71, 0.90)     | 0.83 (0.75, 0.96)     | 0.76 (0.75, 0.78)    |
|            | $f(r, p)$                    | 0.85 (0.77, 0.94)    | 0.82 (0.72, 0.97)     | 0.84 (0.73, 1.00)     | 0.88 (0.85, 0.91)    |
|            | Total Requests               | 10217                | 27436                 | 21416                 | 912                  |
|            | Mean Improvement per Request | $2.54 \times e^{-6}$ | $1.17 \times e^{-6}$  | $6.53 \times e^{-7}$  | $1.31 \times e^{-4}$ |
|            | Runtime (sec)                | 2877                 | 3803                  | 726                   | 663                  |
| 0.1        | $f(r, p)$                    | 0.83 (0.76, 0.93)    | 0.79 (0.70, 0.93)     | 0.79 (0.66, 0.99)     | 0.86 (0.84, 0.89)    |
|            | Total Requests               | 10217                | 28890                 | 22920                 | 982                  |
|            | Mean Improvement per Request | $4.89 \times e^{-7}$ | $-5.54 \times e^{-7}$ | $-4.10 \times e^{-6}$ | $1.01 \times e^{-4}$ |
|            | Runtime (sec)                | 2913                 | 3963                  | 695                   | 647                  |
| 0.2        | $f(r, p)$                    | 0.82 (0.75, 0.91)    | 0.77 (0.68, 0.91)     | 0.73 (0.59, 0.98)     | 0.84 (0.81, 0.86)    |
|            | Total Requests               | 10217                | 30426                 | 23968                 | 1280                 |
|            | Mean Improvement per Request | $4.89 \times e^{-7}$ | $-5.25 \times e^{-7}$ | $-3.92 \times e^{-6}$ | $6.25 \times e^{-5}$ |
|            | Runtime (sec)                | 2930                 | 3912                  | 713                   | 616                  |
| 0.3        | $f(r, p)$                    | 0.81 (0.74, 0.90)    | 0.76 (0.68, 0.89)     | 0.68 (0.52, 0.98)     | 0.81 (0.78, 0.84)    |
|            | Total Requests               | 10217                | 31504                 | 25580                 | 1262                 |
|            | Mean Improvement per Request | 0.00                 | $-7.94 \times e^{-7}$ | $-5.94 \times e^{-6}$ | $3.96 \times e^{-5}$ |
|            | Runtime (sec)                | 2959                 | 3874                  | 618                   | 748                  |

too. Even though, mean improvements per request of ICHEA are still higher than that of other systems under all three error rate cases. ICHEA is the only one who still outperforms its non-interactive version under the case with an error rate of 0.3. Apart from that, the runtime needed by ICHEA is less than other matching systems. Therefore, for large-scale ontology matching problem, ICHEA also efficiently makes use of user validation to obtain high-quality ontology alignment.

### 9.3. Statistical Comparison

In this section, the statistical comparison among ICHEA and other ontology matchers is presented. The values shown in Table 8 on interactive tracks are the mean values under various user error rates. The comparison is formally carried out by means of a multiple comparison procedure which consists of two steps. In the first step, a statistical technique, i.e. the Friedman’s test [45], is used to determine whether there are any differences among the results provided by various approaches. Secondly, if differences are detected in Step 1, a post-hoc test, i.e. Holm’s test [46], is carried out to determine which method outperforms the others.

Friedman’s test is a non-parametric statistical procedure which aims to detect the existence of significant differences among the behaviors of two or more algorithms. In particular, under the null-hypothesis, it states that all

Table 8: Friedman’s test on the alignment’s quality. Each value represents the f-measure, and the number in round parentheses is the corresponding computed rank.

| Matching System | Benchmark   | Anatomy    | Conference  | Large Bio   | Average       |
|-----------------|-------------|------------|-------------|-------------|---------------|
| AML             | 0.38 (10)   | 0.94 (1.5) | 0.73 (2)    | 0.82 (2)    | 0.71 (3.87)   |
| CroMatch        | 0.89 (2.5)  | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.22 (8.62)   |
| Lily            | 0.89 (2.5)  | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.22 (8.62)   |
| LogMap          | 0.55 (8)    | 0.88 (4)   | 0.69 (3)    | 0.78 (3)    | 0.72 (4.50)   |
| LogMapLt        | 0.46 (9)    | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.11 (10.25)  |
| PhenoMF         | 0.01 (13)   | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.002 (11.25) |
| PhenoMM         | 0.01 (13)   | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.002 (11.25) |
| PhenoMP         | 0.01 (13)   | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.002 (11.25) |
| XMap            | 0.56 (7)    | 0.89 (3)   | 0.00 (10.5) | 0.00 (10.5) | 0.36 (7.75)   |
| LogMapBio       | 0.32 (11)   | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.08 (10.75)  |
| ALin            | 0.00 (15.5) | 0.78 (5)   | 0.68 (4)    | 0.00 (10.5) | 0.36 (8.75)   |
| ServOMBI        | 0.00 (15.5) | 0.00 (11)  | 0.00 (10.5) | 0.76 (4)    | 0.19 (10.25)  |
| GA              | 0.71 (5)    | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.17 (9.25)   |
| PSO             | 0.68 (6)    | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.17 (9.50)   |
| MA              | 0.79 (4)    | 0.00 (11)  | 0.00 (10.5) | 0.00 (10.5) | 0.19 (9.00)   |
| ICHEA           | 0.90 (1)    | 0.94 (1.5) | 0.77 (1)    | 0.84 (1)    | 0.86 (1.12)   |

algorithms are equivalent, which means a rejection of this hypothesis implies the existence of differences among the performance of all studied algorithms [47]. In order to reject the null hypothesis, the computed value  $\mathcal{X}_r^2$  must be equal to or greater than the tabled critical chisquare value at a specified level of significance [48]. In our experiment, a level of significance  $\alpha = 0.05$  is chosen. Since 16 approaches are compared in our case, our analysis had to consider the critical value  $\mathcal{X}_{0.05}^2$  for 15 degrees of freedom which is equal to 24.996.

In Table 6, the Friedman’s test shows that the computed  $\mathcal{X}_r^2$  value is 141.06, which is greater than its associated critical value  $\mathcal{X}_{0.05}^2 = 24.996$ . Therefore, the null hypothesis is rejected, which implies that there is a significant difference among these proposals. Consequently, a post-hoc statistical analysis is needed to conduct pairwise comparisons in order to detect concrete differences among compared algorithms. Holm’s test works on a family of hypotheses where each one is related to a comparison between the control method and one of the remaining algorithms. The computed  $z$  value is used to find the corresponding probability from the table of the normal distribution (the so-called  $p$ -value), which is then compared with an appropriate level of significance  $\alpha$ . In order to perform its evaluation, Holm’s method

Table 9: Holm's test on the alignment's quality.

| $i$ | approach  | $z$ value | unadjusted $p$ -value | $\frac{\alpha}{k-i}, \alpha = 0.05$ |
|-----|-----------|-----------|-----------------------|-------------------------------------|
| 15  | AML       | 3.87      | 0.040                 | 0.050                               |
| 14  | LogMap    | 4.50      | 0.010                 | 0.025                               |
| 13  | XMap      | 7.75      | $8.50 \times e^{-7}$  | 0.016                               |
| 12  | CroMatch  | 8.62      | $2.69 \times e^{-8}$  | 0.012                               |
| 11  | Lily      | 8.62      | $2.69 \times e^{-8}$  | 0.010                               |
| 10  | ALin      | 8.75      | $1.51 \times e^{-8}$  | 0.008                               |
| 9   | MA        | 9.00      | $4.35 \times e^{-9}$  | 0.007                               |
| 8   | GA        | 9.25      | $1.63 \times e^{-9}$  | 0.006                               |
| 7   | PSO       | 9.50      | $4.97 \times e^{-10}$ | 0.005                               |
| 6   | LogMapLt  | 10.25     | $1.20 \times e^{-11}$ | 0.005                               |
| 5   | ServOMBI  | 10.25     | $1.20 \times e^{-11}$ | 0.004                               |
| 4   | LogMapBio | 10.75     | $8.67 \times e^{-13}$ | 0.004                               |
| 3   | PhenoMF   | 11.25     | $5.48 \times e^{-14}$ | 0.003                               |
| 2   | PhenoMM   | 11.25     | $5.48 \times e^{-14}$ | 0.003                               |
| 1   | PhenoMP   | 11.25     | $5.48 \times e^{-14}$ | 0.003                               |

sequentially checks the hypotheses ordered by their significance. In details, given the number of compared algorithms  $k$ , it orders the  $p$ -values by denoting them as  $p_1, p_2, \dots, p_{k-1}$  so that  $p_1 \leq p_2 \leq \dots \leq p_{k-1}$ . Then, it compares each  $p_i$  with  $\frac{\alpha}{k-i}$  starting from the most significant  $p$ . If  $p_1$  is below  $\frac{\alpha}{k-1}$ , the corresponding hypothesis is rejected and we are allowed to compare  $p_2$  with  $\frac{\alpha}{k-2}$ . If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis cannot be rejected, all the remaining hypotheses are retained as well.

Holm's test works by setting a control algorithm, and comparing it with the remaining ones. Normally, the algorithm which obtains the lowest value of ranking in the Friedman's test will be chosen as the control algorithm. In this experiment, as is shown in Table 8, ICHEA is characterized by the lowest value of ranking. Under the circumstance of  $\alpha = 0.05$  in this experiment, the results of Holm's test are shown in Table 9. Data analysis of Table 9 depicts that our proposal statistically outperforms other approaches on f-measure at 5% significance level.

#### 9.4. *Lessons Learned and Discussion*

In ICHEA, the user gets involved in the evolving process of the algorithm. In the meantime, user validation is efficiently utilized to update the PRA and guide the searching direction for the algorithm. In particular, the user is asked for feedbacks on some iterations during the matching process. In each iteration, the alignment from the previous iteration is further improved, which enhances the quality of the ontology alignment in a more effective way. On average, the evolutionary process gets stuck (when user gets involved) for 5 times in Benchmark track, 8 times in Conference track, 19 times in Anatomy track, and 41 times, in Large Bio track during the segment pair matching process. Comparing with other interactive ontology matching systems, which got user involved before the matching process (to fine tune the system parameters) or after the matching process (to filter the final alignment), ICHEA benefited more from users' knowledge. Last but not least, the proposed asymmetrical profile-based similarity measure and mapping confidence propagation not only increased the value of user involvements but also reduced the negative effects brought by the erroneous user validations, which further ensured the efficiency of ICHEA. Therefore, ICHEA can more efficiently determine high-quality ontology alignment than other state-of-the-art interactive ontology matching systems.

ICHEA combines the mechanisms of a classic EA with competitive learning, which achieves better and faster results than classic EA does. The unique competitive learning helps ICHEA to achieve good results in terms of both solution quality and runtime, which turns out to be effective in determining the optimal solution. Moreover, it does not require all the mechanisms of an EA, but only a few steps in the algorithm, which are small and simple. It further introduces user involvement (global search) and local refinements (local search), which allows a high level of diversity in solutions, thus reducing the possibility of the premature convergence. It also increases the convergence speed via the local search which in fact, greatly improves the quality of the solution and enables the solution to approach the optimal solution more quickly. In scenarios of complex optimization, the performance of state-of-the-art evolutionary computation techniques, such as GA, MA and PSO, depends largely on the adequate setting of parameters like the probabilities of cross and mutation, size of the population and rate of generational reproduction. Their performance is poor when the designed operators can not guarantee the correct movements of the populations in the search space. Comparing with state-of-the-art evolutionary computation techniques, ICHEA works upon



the probabilistic modeling of promising solutions, which makes it easier to predict the movements of the populations in the search space, as well as to avoid the need for many parameters. In addition, huge memory consumption seriously limits the application of population-based algorithm. In this experiment, population-based algorithms consumed about 150 gigabytes on average to determine the alignments for small-scale ontology matching tasks in Benchmark. Due to memory overflow, they can not effectively determine the alignments for large-scale ontology matching tasks in Conference, Anatomy and Large Bio tracks. However, due to the ontology partition, compact encoding mechanism and efficient user involvement, ICHEA can efficiently deal with various large-scale ontology matching tasks. To conclude, ICHEA outperforms other EA-based approaches in terms of both quality of alignments and computation efficiency.

## 10. Conclusion and Future Work

Nowadays, how to efficiently make use of the user validation to improve the ontology alignment still retains a challenge in the interactive ontology matching domain. To address this challenge, we present a PRA-based interactive ontology matching technique, which gets the user involved into a PRA-based iterative automatic-matching process to balance between user workload and the demand of improving the quality of ontology alignment. Particularly, ICHEA proposes to adaptively determine the timing of getting the user involved, and present the user with the most problematic mappings for validation, and help him to validate multiple conflicting mappings simultaneously. Meanwhile, it propagates the user validations to maximize their value and decrease the negative effect brought by erroneous user validations. The well-known OAEI 2016's benchmark track and interactive track are utilized to test ICHEA's performance. Results of the experiment on benchmark track show that both the f-measure and the f-measure provided per second of ICHEA outperform those of all the OAEI participants and three state-of-the-art EA-based ontology matching techniques. In addition, results of the experiment on three interactive testing cases further show that ICHEA can efficiently determine high-quality ontology alignments under different error rate cases, and the performance of ICHEA in general is better than that of state-of-the-art interactive ontology matching systems.

This paper utilizes the oracle to simulate the human expert's behaviors in the experiment. However, in real applications, manual validations by human

experts need to be done within a limited time frame, where errors are prone to occur, because of different levels of expertise, misinterpretations, human biases, etc. Therefore, in the future, new techniques will be investigated to help human experts to make correct decisions in the manually validating process more efficiently. To achieve this, the first step is to add a graphical user interface, which aims to solve the problem like how to visually present the ontology mappings and ontology information, and enable easy retrieve to release the user from memorizing information. Basically, the interactions done by the user in this work is either to accept or reject a certain mapping suggestion. It may also happen that a mapping is not suggested but is needed according to the user. In this case, additional functionalities are proposed to manually add on a mapping. Moreover, it is also interesting to explore how to alert the user when contradicting validations are made, or preventively remove mappings that lead to logical conflicts. Last but not least, other evolutionary methodology can be tried, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [49], to implement an automatic matching procedure to further improve the performance.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61503082 and 61329302), Natural Science Foundation of Fujian Province (No. 2016J05145), Scientific Research Startup Foundation of Fujian University of Technology (No. GY-Z15007), Scientific Research Development Foundation of Fujian University of Technology (No. GY-Z17162), Fujian Province Outstanding Young Scientific Researcher Training Project (No. GY-Z160149) and China Scholarship Council. Xin Yao was supported by a Royal Society Wolfson Research Merit Award.

## References

- [1] X. Xue, J. Liu, Optimizing Ontology Alignment Through Compact MOEA/D, *International Journal of Pattern Recognition and Artificial Intelligence* 31 (4) (2017) 1–20.
- [2] P. Shvaiko, J. Euzenat, Ontology matching: state of the art and future challenges, *IEEE Transactions on knowledge and data engineering* 25 (1) (2013) 158–176.

- [3] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, C. Pesquita, User validation in ontology alignment, in: International Semantic Web Conference, Springer, 200–217, 2016.
- [4] J. Kenny, Particle swarm optimization, in: Proc. 1995 IEEE Int. Conf. Neural Networks, 1942–1948, 1995.
- [5] N. Krasnogor, J. Smith, A memetic algorithm with self-adaptive local search: TSP as a case study, in: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., 987–994, 2000.
- [6] P. Merz, B. Freisleben, Memetic algorithms for the traveling salesman problem, *Complex Systems* 13 (4) (2001) 297–346.
- [7] G. Acampora, V. Loia, A. Vitiello, Enhancing ontology alignment through a memetic aggregation of similarity measures, *Information Sciences* 250 (2013) 1–20.
- [8] X. Xue, Y. Wang, Optimizing Ontology Alignments through a Memetic Algorithm Using both MatchFmeasure and Unanimous Improvement Ratio, *Artificial Intelligence* 223 (2015) 65–81.
- [9] X. Xue, Y. Wang, Using Memetic Algorithm for Instance Coreference Resolution, *IEEE Transactions on Knowledge and Data Engineering* 28 (2) (2016) 580–591.
- [10] K. E. Parsopoulos, Cooperative micro-differential evolution for high-dimensional problems, in: Proceedings of the conference on Genetic and evolutionary computation, Montreal, Canada, 531–538, 2009.
- [11] F. Neri, G. Iacca, E. Mininno, Compact optimization, vol. 38, Springer, Berlin, Germany, 337–364, 2013.
- [12] P. Lambrix, Q. Liu, Using partial reference alignments to align ontologies, in: European Semantic Web Conference, Springer, 188–202, 2009.
- [13] X. Xue, Y. Wang, A. Ren, Optimizing ontology alignment through memetic algorithm based on partial reference alignment, *Expert Systems with Applications* 41 (7) (2014) 3213–3222.

- [14] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, A. Gómez-Rodríguez, Ontology matching: A literature review, *Expert Systems with Applications* 42 (2) (2015) 949–971.
- [15] D. Faria, C. Pesquita, B. S. Balasubramani, C. Martins, J. Cardoso, H. Curado, F. M. Couto, I. F. Cruz, OAEI 2016 Results of AML, *Ontology Matching* (2016) 138–145.
- [16] E. Jiménez-Ruiz, B. C. Grau, V. Cross, LogMap family participation in the OAEI 2016, *Ontology Matching* (2016) 185–189.
- [17] D. Warith, Eddine, K. Mohamed, Tarek, Y. Sadok, BEN, XMap: results for OAEI 2016, *Ontology Matching* (2016) 222–226.
- [18] N. Kheder, G. Diallo, ServOMBI at OAEI 2015, in: *OM*, 200–207, 2015.
- [19] G. Diallo, Efficient building of local repository of distributed ontologies, in: *Signal-Image Technology and Internet-Based Systems (SITIS)*, 2011 Seventh International Conference on, IEEE, 159–166, 2011.
- [20] P. Lambrix, R. Kaliyaperumal, A session-based approach for aligning large ontologies, in: *Extended Semantic Web Conference*, Springer, 46–60, 2013.
- [21] S. Duan, A. Fokoue, K. Srinivas, One size does not fit all: Customizing ontology alignment using user feedback, in: *International Semantic Web Conference*, Springer, 177–192, 2010.
- [22] H. Tan, P. Lambrix, A method for recommending ontology alignment strategies, in: *The Semantic Web*, Springer, 494–507, 2007.
- [23] D. Ritze, H. Paulheim, Towards an automatic parameterization of ontology matching tools based on example mappings, in: *Proceedings of the 6th International Conference on Ontology Matching*, vol. 814, CEUR-WS. org, 37–48, 2011.
- [24] C. J. V. Rijsberge, *Information Retrieval*, University of Glasgow, Butterworth, London, 1975.
- [25] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, C. Trojahn, Ontology alignment evaluation initiative: six years of experience, in: *Journal on data semantics XV*, Springer, 158–192, 2011.

- [26] G. Acampora, V. Loia, A. Vitiello, Enhancing ontology alignment through a memetic aggregation of similarity measures, *Information Sciences* 250 (2013) 1–20.
- [27] P. Bentley, D. Corne, *Creative evolutionary systems*, Morgan Kaufmann, 2002.
- [28] E. Rahm, Towards large-scale schema and ontology matching, *Schema Matching and Mapping* (2011) 3–27.
- [29] X. Xue, S.-C. Chu, An Alignment-Oriented Segmenting Approach for Optimizing Large Scale Ontology Alignments, *Journal of Internet Technology* 17 (7) (2016) 1373–1382.
- [30] N. Yuruk, X. X. M. Mete, T. A. J. Schweiger, AHSCAN: Agglomerative hierarchical structural clustering algorithm for networks, in: *International Conference on Advances in Social Network Analysis and Mining*, Athens, Greece, 72–77, 2009.
- [31] M. S. Falconer, F. N. Noy, Interactive techniques to support ontology matching, in: *Schema Matching and Mapping*, Springer, 29–51, 2011.
- [32] A. Maedche, S. Staab, Measuring Similarity between Ontologies, in: *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management*, Ischia Island, Italy, 251–263, 2002.
- [33] G. Stoilos, G. Stamou, S. Kollias, A string metric for ontology alignment, in: *Proceedings of 4th International Semantic Web Conference (ISWC 2005)*, Galway, Ireland, 623–637, 2005.
- [34] G. A. Miller, WordNet: A lexical database for English, *Communications of the ACM* 38 (11) (1995) 39–41.
- [35] W. Winkler, The state record linkage and current research problems, *Tech. Rep. RR99-04*, Statistics of Income Division, Washington DC, USA, 1999.
- [36] P. Wang, Y. Zhou, B. Xu, Matching large ontologies based on reduction anchors, in: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2343–2348, 2011.

- [37] M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, et al., Results of the ontology alignment evaluation initiative 2016, in: OM: Ontology Matching, No commercial editor., 73–129, 2016.
- [38] J. Martinez-Gil, E. Alba, J. F. A. Montes, Optimizing ontology alignments by using genetic algorithms, in: Proceedings of the First International Conference on Nature Inspired Reasoning for the Semantic Web-Volume 419, CEUR-WS. org, 1–15, 2008.
- [39] J. Bock, J. Hettenhausen, Discrete particle swarm optimisation for ontology alignment, *Information Sciences* 192 (2012) 152–173.
- [40] M. Gulić, B. Vrdoljak, M. Banek, CroMatcher results for OAEI 2016, *Ontology Matching* (2016) 153–160.
- [41] W. Wang, P. Wang, Lily results for OAEI 2016, *Ontology Matching* (2016) 178–184.
- [42] O. Bodenreider, The unified medical language system (UMLS): integrating biomedical terminology, *Nucleic acids research* 32 (suppl 1) (2004) D267–D270.
- [43] R. G. Miguel, Ange, G. P. N. S. R. H. Georgios, V, Integrating phenotype ontologies with PhenomeNET, *Ontology Matching* (2016) 201–209.
- [44] R. Hoehndorf, P. N. Schofield, G. V. Gkoutos, PhenomeNET: a whole-phenome approach to disease gene discovery, *Nucleic acids research* 39 (18) (2011) e119–e119.
- [45] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *Journal of the american statistical association* 32 (200) (1937) 675–701.
- [46] S. Holm, A simple sequentially rejective multiple test procedure, *Scandinavian journal of statistics* (1979) 65–70.
- [47] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 special session on real parameter optimization, *Journal of Heuristics* 15 (6) (2009) 617–644.

- [48] D. J. Sheskin, Handbook of parametric and nonparametric statistical procedures, crc Press, 2003.
- [49] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary computation 9 (2) (2001) 159–195.